

A Probabilistic Approach to Robust Execution of Temporal Plans with Uncertainty

Ioannis Tsamardinis

Department of Biomedical Informatics, Vanderbilt University,
Eskind Biomedical Library, Room 444,
2209 Galrand Ave, Nashville, TN 37232-8340,
USA
ioannis.tsamardinis@vanderbilt.edu

Abstract.

In Temporal Planning a typical assumption is that the agent controls the execution time of all events such as starting and ending actions. In real domains however, this assumption is commonly violated and certain events are beyond the direct control of the plan's executive. Previous work on reasoning with uncontrollable events (Simple Temporal Problem with Uncertainty) assumes that we can bound the occurrence of each uncontrollable within a time interval. In principle however, there is no such bounding interval since there is always a non-zero probability the event will occur outside the bounds. Here we develop a new more general formalism called the Probabilistic Simple Temporal Problem (PSTP) following a probabilistic approach. We present a method for scheduling a PSTP maximizing the probability of correct execution. Subsequently, we use this method to solve the problem of finding an optimal execution strategy, i.e. a dynamic schedule where scheduling decisions can be made on-line.

1. Introduction

Planning is a major area of Artificial Intelligence researching the following problem: given a description of available actions and desired goals, discover a course of action that achieves the goals. This is a notoriously hard problem in general and Classical Planning [1] was developed as a set of assumptions, restrictions, formalisms, and representations to solve it. Classical planning assumes actions are instantaneous and thus there is no need to represent time explicitly. Temporal Planning is the effort of enriching Classical Planning with an explicit representation of time and dealing with durative actions and temporal constraints. Recently, there is a revived interest in temporal planners [2, 3], because of the successes in new, interesting, and real-world situated domains (e.g. Remote Agent [2]). A temporal planning competition in the next Artificial Intelligence Planning and Scheduling conference has also been announced.

Temporal planners are typically based on temporal reasoning formalisms such as the Simple Temporal Problem (STP) [4]. Most often, a planning action A is represented in an STP with two temporal variables $start(A)$ and $end(A)$ corresponding to the instantaneous events of starting and ending A respectively. Then, constraints on the variables can be defined. The STP and other related formalisms are able to answer the question whether a set of constraints is consistent. *Consistency* is defined as the existence of an assignment to the temporal variables that respects the constraints.

Even though the STP allows the construction of temporal planners that significantly extend Classical Planning's capabilities, it still makes the unrealistic assumption that all events are controllable, i.e. under the direct control of the executing agent, and thus the latter can assign any time it desires to the variables in order to execute the plan in a way that respects all the constraints. In the real world however, certain events are *uncontrollable*; it is Nature¹ that determines their exact timing. For example, the event corresponding to $end(A)$, where A is the action of driving a truck from one place to another is uncontrollable. The planning agent can set the time for $start(A)$, i.e. the time the truck will leave its origin, but it is traffic conditions that determine the exact time assigned to $end(A)$, i.e. the time the truck will arrive to its destination.

The first formalism that explicitly models uncontrollable events is the STPU (STP with Uncertainty) [5]. In an STPU, the uncontrollable events are presumed to occur within certain bounds. A new concept in place of consistency is introduced; that of *controllability*. In particular, if there is an assignment to the controllables that respects the constraints no matter what values (within the bounds) Nature selects for the uncontrollables, the STPU is *strongly consistent* (two other related concepts, *dynamic* and *weak controllability* are also defined for STPUs).

The STPU formalism can be used to model the uncertainty of the timing of events in temporal plans; nevertheless, it has certain serious drawbacks. The first is that *in principle there are no such bounds*: there is always a non-zero probability that exogenous factors will cause an event to occur some time outside the bounds. The second problem is that *there is no principled way of selecting the bounds according to some optimality criterion*. The person who encodes the domain is responsible for the difficult task of choosing reasonable approximate bounds for such constraints. If these bounds are loose, Nature will likely respect them, but it will be less probable that the STPU will be strongly consistent. In the latter case, STPU algorithms provide no help as to how to execute the plan. If, on the other hand, the bounds are too tight, it might be easier to find assignments that respect the constraints no matter the values of the uncontrollables, yet less likely for Nature to respect the defined bounds.

In this paper, we take a new and more general, probabilistic approach to the problem of executing temporal plans with uncertainty in which we model the uncontrollable events using random variables following conditional probability distributions. Our approach does not have the above two STPU drawbacks. It also subsumes STPUs since each STPU uncontrollable variable constrained to occur within a lower and upper bound is equivalent in our approach with a random variable following a uniform distribution within these bounds.

To represent such probabilistic information, we develop a new temporal reasoning formalism that we call *Probabilistic Simple Temporal Problem* (PSTP). In STPUs we

¹ By Nature we will call all exogenous factors outside the agent's control.

seek for an assignment to the controllables that respects the constraints in all cases, assuming bounds on the contingent links. The problem in PSTPs is, instead, to find the assignment that *maximizes the probability* that Nature will select values that will respect all constraints. We subsequently show how to solve this optimization problem for the class of PSTPs in which the probability distributions are time-invariant, there are no constraints between two uncontrollable events, and the probability distributions are continuous. The new formalism can be used to robustly execute temporal plans with uncontrollable events. We also present an algorithm for dynamically executing a temporal plan while maximizing our chances of executing it while respecting the constraints.

The structure of the paper is as follows: Section 2 presents the necessary background information; Section 3 defines the PSTP while Section 4 solves the problem of maximizing the probability of successful execution when committing to a specific schedule. Section 5 deals with the problem of executing PSTPs and dynamically deciding the execution time of the controllable events. Sections 6 and 7 provide with a discussion and conclusion to the paper.

2. Background

Definition 1: A *Simple Temporal Problem* (STP) is a tuple $\langle V, E \rangle$, where V is a set of temporal variables (also called nodes, time-points, and events) and E a set of constraints of the form $l \leq x - y \leq u$, where $x, y \in V$ and $l, u \in \mathcal{R}$. A *Simple Temporal Problem with Uncertainty* (STPU) is a tuple $\langle V, E, C \rangle$ where V, E are defined as in the STP case, and C is a set of contingent constraints of the form $l \leq x - y \leq u$, $x, y \in V$ where x is an uncontrollable variable. The semantics of contingent constraints is that Nature will select a time for x in the interval $[y+l, y+u]$.

Because STPs and STPUs contain only binary constraints (i.e. involving only two variables) we can represent them with graphs where the nodes correspond to the variables and the edges correspond to the constraints. To be able to express unary constraints of the form $l \leq x \leq u$ we define a new variable TR (time reference) that is associated with an arbitrary point in time that is considered the time zero. Then, unary constraints $l \leq x \leq u$ are expressed as binary constraints $l \leq x - TR \leq u$. The graph of an example STP that expresses a simple plan to complete a surgical operation is shown in Figure 1(a). The operation will last between 20' to 35' minutes and we would like to exit the operating room at most 5' late for the next operation or at most 10' early. The next operation is set to take place any time between 8:00 and 9:00 in the morning.

The set of time-points V is the set $\{TR, Operation-Start, Operation-End, Next-Operation-Start\}$ with obvious semantics that we will abbreviate as $\{TR, OS, OE, NOS\}$. The set of constraints E corresponds to the set of edges; each edge from y to x is annotated with an interval $[l, u]$ and expresses the constraint $l \leq x - y \leq u$. E.g. the edge from OS to OE annotated with the interval $[20', 35']$ expresses the constraint $20' \leq OE - OS \leq 35'$ that can be read as “ OE follows OS by at least 20' minutes and at most 35' minutes”. Similarly, the edge from OE to NOS annotated with the interval $[-5', 10']$ corresponds to the constraint $-5' \leq NOS - OE \leq 10'$, i.e. the fact that we finish the first

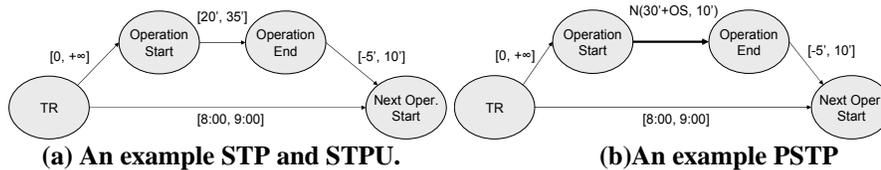


Figure 1

operation at most 5' late and at most 10' early. An STP is *consistent* if there is an assignment to the variables that respects the constraints. The example of Figure 1(a) is consistent since the assignment $\{TR \leftarrow 0, OS \leftarrow 7:30, OE \leftarrow 8:00, NOS \leftarrow 8:00\}$ satisfies all constraints. There are a number of polynomial time algorithms that determines consistency in STPs [6].

As already mentioned, STPs do not explicitly distinguish between controllable and uncontrollable variables. In the above example, OE is an uncontrollable variable since the doctor cannot predict or determine the exact moment the operation will end. Thus consistency does not guarantee correct execution. The STPU performs this distinction by defining contingent constraints. The graph of Figure 1(a) can be thought of as expressing an STPU if we define that $\{20' \leq OE - NOS \leq 35'\}$ to be a contingent constraint.

The question that arises is whether it is possible to execute such an STPU in a way that respects the constraints in which case we will call the STPU *controllable*. We distinguish the following cases: If there is a single assignment that respects the constraints no matter when the uncontrollables occur then the STPU is *strongly controllable*. In this case we can schedule the controllables without any knowledge about the uncontrollables. A different case is when it is possible to schedule the controllables if we are given full knowledge about the times of uncontrollables. Then the STPU is *weakly controllable*. Finally, if it is possible to schedule the controllables dynamically (on-line) by using information of only the uncontrollables observed thus far, then the STPU is *dynamically controllable*. Strong controllability is polynomial, while weak controllability is co-NP-complete [7]. A recent important result is that dynamic controllability can be determined in polynomial time [5].

The STPU of Figure 1(a) is strongly consistent because the assignment $\{TR \leftarrow 0, OS \leftarrow 7:30, NOS \leftarrow 8:00\}$ respects the constraints no matter when OE occurs. To see this take the extreme situations for OE : if we finish after only 20', time will be 7:50, which is no more than 10' early. If we finish after 35', then we will be done at 8:05, which is no more than 5' late. Notice however, that if the bounds on the contingent link were not $[20', 35']$ but looser, e.g. $[10', 40']$, then the STPU would not be strongly controllable. In this case, a controllability-checking algorithm would only return a "no" answer without any further information as to how to execute the STPU. In addition, there is no principled way to select the bounds $[20', 35']$ since in general there is always a non-zero probability a value outside these bounds will be selected by Nature. We now define the PSTP that deals with these problems by employing a probabilistic representation.

3. The Probabilistic Simple Temporal Problem

Definition 2: A *Probabilistic Simple Temporal Problem* (PSTP) is a tuple $\langle V_1, V_2, E, S, Pa \rangle$ where:

V_1 is a set of controllable variables (interchangeably also called events, nodes, and time-points) each with domain the set of reals \mathfrak{R}

V_2 is a set of uncontrollable random variables each with domain the set of reals \mathfrak{R}

E a set of constraints of the form $x - y \leq b_{yx}$ where $x, y \in V_1 \cup V_2$ and $b_{yx} \in \mathfrak{R} \cup \{-\infty, +\infty\}$

Pa a function $V_2 \rightarrow V_1$ providing for each uncontrollable x , its parent $Pa(x) = y \in V_1$

S a set of conditional probability density functions (pdf) $p(x \mid Pa(x) = t)$ for each $x \in V_2$

Notice that the constraints in a PSTP have the form $x - y \leq b_{yx}$ instead of $l \leq x - y \leq u$ as in the STP/STPU case. However, the two definitions are equivalent since we can represent the latter with the former and vice versa: $(l \leq x - y \leq u) \Leftrightarrow (x - y \leq u) \wedge (y - x \leq -l)$. We preferred the definition using a single inequality because it is more convenient for algebraic manipulations. Also notice that each pdf p in a PSTP defines a corresponding cumulative probability function (cdf) P .

An example PSTP is shown in Figure 1(b) where the same plan as in the STP of is depicted. The set of controllable variables V_1 is the set $\{TR, OS, NOS\}$ while the set of uncontrollable variables V_2 is the set $\{OE\}$. The pdf for the only uncontrollable event OE is shown with the bold edge in the figure denoting that the parent of OE is OS ($Pa(OE) = OS$) and that $p(OE \mid OS) = N(OS + 30', 10')$, where $N(\mu, \sigma)$ is a Gaussian (normal) density function with mean μ and standard deviation σ . In other words, if we start operating at some time OS , we expect to finish, on average, in 30 minutes. The constraints E correspond to the non-bold edges in the figure. Notice that, even though it is the end of actions that are uncontrollable events in the example, this is not a general restriction of PSTPs and any event can be defined uncontrollable.

The problem now becomes *executing a PSTP so that the probability that all constraints are satisfied is maximized*. Similar to the STPU case the answer to this problem depends on the assumptions we make regarding our knowledge about the uncontrollable variables. We distinguish between two cases:

- The controllables have to be pre-scheduled (off-line) before execution begins, i.e. at scheduling time we have no knowledge about the uncontrollables during execution. This is the case for example, if the scheduling algorithm is not part of the executive of the plan. We will call this problem the *Static Scheduling Optimization Problem* and it corresponds to finding a *strong execution strategy* in STPUs.
- We schedule the controllables dynamically (on-line) using information about the uncontrollables that have already been observed and the information that some have not been observed yet. This is the most typical case in planning and corresponds to finding a *dynamic execution strategy* in STPUs. We will call it *Dynamic Scheduling Optimization Problem*.

The above two cases describe two optimization problems that arise when executing a temporal plan represented as a PSTP. We now turn our focus to the static scheduling optimization problem.

4. Static Scheduling Optimization

Let us denote with $P(\text{Success}|T)$ the probability that all constraints will be satisfied during execution if the controllables are executed according to schedule T . Then, the static optimization problem can be cast as the following mathematical program:

$$\begin{aligned} & \text{maximize } P(\text{Success}|T), \\ & \text{subject to:} \\ & \quad y - z \leq b_{zy}, y, z \in V_1, b_{zy} \in \mathfrak{R} \end{aligned}$$

We will see that it is actually easier to maximize the logarithm of the objective function. Later in this section we derive an analytical expression for $\log P(\text{Success}|T)$ under certain assumptions about the PSTP, from which it becomes obvious that $\log P(\text{Success}|T)$ is not linear. Therefore, the optimization problem is a non-linear constrained optimization problem with linear constraints. There are several methods for solving such problems in the literature and can be broadly categorized in terms of the derivative information, i.e. whether it is available or not. If it is not available then only function evaluations can be used for finding an optimum. If the derivative is available then a search can be performed in the direction of the steepest ascent $\nabla P(\text{Success}|T)$ (gradient ascent). Using the gradient is more efficient in general when the derivative is continuous. With constrained optimization one has to pay attention to the constraints so that search remains in the feasible region during gradient ascent.

Later in this section we provide analytical expressions for both the function and the derivative. One can then use standard optimization solvers such as the one included with the software package MATLAB; the latter uses a version of the Sequential Quadratic Programming (SQP) described in [8]. Most of these methods however, do not guarantee finding the global optimum and only discover local optima.

In order to be able to derive analytical expression and use the SQP method we make the following *assumptions*:

1. All pdf $p(x|Pa(x)=t)$ are time-invariant, meaning that their shape does not depend on the particular value t of $Pa(x)$ (the value of $Pa(x)$ may only translate them on the x -axis). Thus, there is a pdf $p'(x)$ for which $p(x|Pa(x)=t)=p'(x+t)$. If P and P' are the cdf corresponding to the pdf p and p' respectively, then $P(x \leq b | Pa(x)=t) = P'(x \leq b-t)$. Notice that the pdf $N(30'+LH, '10)$ of Figure 1(b) satisfies this assumption. The shape of the density function does not depend on the exact value of OS : we will finish the operation in half an hour, on average, no matter when we start operating.
2. There are no constraints among two variables in V_2 , i.e. there are no constraints $l \leq x - y \leq u$ where $x, y \in V_2$.
3. The pdf are continuous.

Let us first derive $P(\text{Success}|T)$ for the case when there is a single uncontrollable variable x in V_2 . Variables y_i will denote the controllables in V_1 .

$$\begin{aligned}
P(\text{Success} | T) &= P(\bigwedge_{ij} z_i - z_j \leq b_{z_j z_i} | T), z_i, z_j \in V_1 \cup V_2 \\
&= P((\bigwedge_j y_j - x \leq b_{xy_j})(\bigwedge_i x - y_i \leq b_{y_i x})(\bigwedge_{ij} y_i - y_j \leq b_{y_j y_i}) | T)
\end{aligned} \tag{1}$$

The equation above means that the probability of *Success* given a schedule T is the probability that all the constraints of the form $x - y \leq b$ are satisfied, all the constraints of the form $y - x \leq b$ are satisfied, *and* all the constraints $y_i - y_j \leq b$ between the controllables are satisfied. Since we perform a search in the space of feasible schedules T , all constraints among the controllables are satisfied and so the above equation becomes:

$$P(\text{Success} | T) = P((\bigwedge_j y_j - x \leq b_{xy_j})(\bigwedge_i x - y_i \leq b_{y_i x}) | T) \tag{2}$$

which can be simplified to:

$$\begin{aligned}
P(\text{Success} | T) &= P((\bigwedge_j y_j - b_{xy_j} \leq x)(\bigwedge_i x \leq y_i + b_{y_i x}) | T) \\
&= P(\max_j (y_j - b_{xy_j}) \leq x \wedge x \leq \min_i (y_i + b_{y_i x}) | T) \\
&= P(L_T(x) \leq x \leq U_T(x) | T)
\end{aligned} \tag{3}$$

where $U_T(x) = \min_i (y_i + b_{y_i x})$ and $L_T(x) = \max_j (y_j - b_{xy_j})$ for the given schedule T . When there are more than one uncontrollables x_i then the above equation becomes:

$$P(\text{Success} | T) = P(\bigwedge_i L_T(x_i) \leq x_i \leq U_T(x_i) | T) \tag{4}$$

Note that (i) the random variables x_i are independent of each other (their probability distribution only depends on the value of their parent) and (ii) each conjunction, by assumption (2) contains only one random variable (i.e. it can never be the case that an uncontrollable x_i may appear in $U(x_j)$ or $L(x_j)$). So, the probability of the conjunction of the constraints being satisfied is the product of the probabilities each one of them being satisfied. Using the above facts and maximizing the logarithm of the probability instead we get:

$$\boxed{\log P(\text{Success} | T) = \log P(\bigwedge_i L(x_i) \leq x_i \leq U(x_i) | T) =} \tag{5}$$

$$= \log \prod_i P(L_T(x_i) \leq x_i \leq U_T(x_i) | T) = \sum_i \log P(L_T(x_i) \leq x_i \leq U_T(x_i) | T) = \sum_i \log \{P(x_i \leq U_T(x_i) | T) - P_T(x_i \leq L(x_i) | T)\}$$

Where:

$$P(x_i \leq U(x_i) | T) - P(x_i \leq L(x_i) | T) = \int_{L(x_i)}^{U(x_i)} p(x_i) dx_i \quad (6)$$

Notice that the value of the integral is easily computable for most pdf in the literature. For example, most introductory statistics textbooks contain tables of the function $\Phi(t)$ which is the probability $P(x_i \leq t)$ when x_i follows a normal distribution $N(0, I)$. When x_i follows a normal distribution $N(\mu, \sigma)$ then $P(x_i \leq t)$ is given by the function $\Phi((t-\mu)/\sigma)$.

Equation (5) can be used to calculate $P(\text{Success}|T)$ for any given feasible schedule T when the PSTP satisfies assumption (2). Thus, a search procedure in the space of feasible schedules could be used to optimize this quantity. Nevertheless, such an uninformative search procedure is probably highly inefficient. We now develop analytical expressions for the partial derivatives of $\log P(\text{success}|T)$ when assumption (1) is also satisfied. From Equation (5) and the chain rule for derivatives we get that:

$$\begin{aligned} \frac{\partial \log P(\text{Success} | T)}{\partial y_m} &= \frac{\partial}{\partial y_m} \sum_i \log \{P(x_i \leq U_T(x_i) | T) - P(x_i \leq L_T(x_i) | T)\} = \\ & \sum_i \frac{\partial}{\partial y_m} \log \{P(x_i \leq U_T(x_i) | T) - P(x_i \leq L_T(x_i) | T)\} = \\ & \sum_i \frac{1}{\{P(x_i \leq U_T(x_i) | T) - P(x_i \leq L_T(x_i) | T)\}} \frac{\partial}{\partial y_m} \{P(x_i \leq U_T(x_i) | T) - P(x_i \leq L_T(x_i) | T)\} \end{aligned} \quad (7)$$

The partial derivatives in the above equation can be calculated as follows: first we notice that by assumption (1)

$$\frac{\partial P(x_i \leq U_T(x_i) | T)}{\partial y_m} = \frac{\partial P'(x_i \leq U_T(x_i) - Pa(x) | T)}{\partial y_m} \quad (8)$$

which in turn is:

$$\frac{\partial P'(x_i \leq U_T(x_i) - Pa(x) | T)}{\partial y_m} = p'(U_T(x_i) - T(Pa(x))) \left\{ \frac{\partial U_T(x_i)}{\partial y_m} - \frac{\partial Pa(x)}{\partial y_m} \right\} \quad (9)$$

Notice that by definition of $U_T(x)$, $\partial U_T(x_i)/\partial y_m$ is 1 if y_m is the variable maximizing the quantity $(y_i + b_{y,x})$, otherwise it is 0. Similarly, $\partial Pa(x_i)/\partial y_m$ is 1 if $Pa(x_i)=y_m$ and 0 otherwise. The calculation of $\partial P(x_i \leq L(x_i) | T)/\partial y_m$ is similar and so we finally derive:

$$\frac{\partial \log P(\text{Success} | T)}{\partial y_m} = \sum_i \frac{1}{\{P(x_i \leq U_T(x_i) | T) - P(x_i \leq L_T(x_i) | T)\}} \quad (10)$$

$$(p'(U_T(x_i) - T(Pa(x_i))) \left\{ \frac{\partial U_T(x_i)}{\partial y_m} - \frac{\partial Pa(x_i)}{\partial y_m} \right\} -$$

$$-p'(L_T(x_i) - T(Pa(x_i))) \left\{ \frac{\partial L_T(x_i)}{\partial y_m} - \frac{\partial Pa(x_i)}{\partial y_m} \right\})$$

We will now apply these formulas to the example of Figure 1(b) and for the schedule $T_1 = \{TR \leftarrow 0, OS \leftarrow 7:30, NOS \leftarrow 8:00\}$:

$$P(\text{Success} | T_1) = P(-5' \leq NOS - OE \leq 10' | T_1) = \quad (11)$$

$$P(7:50 \leq OE \leq 8:05 | T_1)$$

Since OE follows the normal distribution $N(30'+OS, 10')=N(8:00, 10')$ the above probability is given by $\Phi((8:05-8:00)/10') - \Phi((7:50-8:00)/10') = 0.53$ or 53%.

The partial derivative of $\log P(\text{Success}|T_1)$ over NOS is given by substitution to Equation (10):

$$\frac{\partial \log P(\text{Success} | T_1)}{\partial NOS} = \frac{1}{P(\text{Success} | T_1)} \quad (12)$$

$$(p'(8:05 - 7:30) \left\{ \frac{\partial U_T(OE)}{\partial NOS} - \frac{\partial Pa(OE)}{\partial NOS} \right\} -$$

$$-p'(7:50 - 7:30) \left\{ \frac{\partial L_T(OE)}{\partial NOS} - \frac{\partial Pa(OE)}{\partial NOS} \right\}) =$$

$$\frac{1}{0.53} (p'(45') - p'(20')) = 1.88(p'(45') - p'(20'))$$

This is because $Pa(OE)=OS$ and so $\frac{\partial Pa(OE)}{\partial NOS} = 0$, while $\frac{\partial U_{T_1}(OE)}{\partial NOS} = \frac{\partial L_{T_1}(OE)}{\partial NOS} = 1$, and also because $U_T(OE)=T(NOS)+b_{NOS,OE} = 8:00 + 5' = 8:05$ and $L_T(OE)=T(NOS)-b_{OE,NOS} = 7:50$. In other words, if schedule the next operation an infinitesimal time $dNOS$ later, we will be adding $p'(45')dNOS$ probability to satisfy the upper bound of the constraint $7:50+dNOS \leq OE \leq 8:05+dNOS$ and subtracting $p'(20')dNOS$ probability to satisfy the

lower bound of this constraint. Because $p'(45') > p'(20')$ for p' being $N(30, 10')$, by scheduling the next operation for a little later than 8:00 we increase our chances of executing the plan correctly. The reader is encouraged to work out the partial derivative $\frac{\partial \log P(\text{Success} | T_i)}{\partial OS}$ and notice that it is equal to $-\frac{\partial \log P(\text{Success} | T_i)}{\partial NOS}$ (the formula is the same, but now $\partial PA(OE) / \partial OS = 1$ and also $\partial U_T(OE) / \partial OS = 0$). That is, we get the same change in probability of success by starting the operation a little earlier instead of scheduling the next one a little later, as we would expect by symmetry. For this simple example, it is easy to see that an optimal solution is the schedule $\{TR \leftarrow 0, OS \leftarrow 7:27:30, NOS \leftarrow 8:00\}$. There is actually an infinite number of optimal solutions where the next operation is scheduled for a time $t \in [8:00, 9:00]$ and OS is set to $t - 0:32:30$, i.e. thirty two and a half minutes earlier.

5. Dynamic Optimization

Unfortunately, deriving analytical expressions for the function and its derivative for the dynamic case is much harder. If we are allowed to use information from observations of the uncontrollables to decide what and when to schedule next, then we are not looking any more for a single optimizing schedule T ; instead we are looking for an execution strategy (policy) St that maps the current history of the system, i.e. our decisions about the controllables and our observations about the uncontrollables to the next decision of what and when to execute next.

Nevertheless, a sketch algorithm to the problem would be to find an initial optimal static schedule T and then loop on the steps below:

1. If it is time to execute a controllable according to T , execute it.
2. If an uncontrollable x has occurred at time t , remove it from V_2 , remove its pdf from the set of pdf S , insert the constraint $t \leq x - TR \leq t$ to the set of constraints E , and update (recalculate) the optimal T .
3. If an uncontrollable x has not occurred by time t' , update its cdf $P(x \leq t' | Pa(x))$ with the posterior $P(x \leq t' | Pa(x), x > t)$, and update the optimal T .

In step (2) whenever we observe an uncontrollable, then we take into consideration this observation and recalculate the optimal schedule T . Not observing an uncontrollable also provides with useful information since it changes the posterior probability of the occurrence of the uncontrollable and should also be taken into account (Step (3)).

Unfortunately, even though the above algorithm constructs a PSTP schedule dynamically, it does not answer the question what is $P(\text{Success} | St)$ where St is a dynamic execution strategy. The latter is important for planning purposes since a planner might decide to keep searching if this probability is below a given threshold or accept a plan if it is above this threshold.

6. Discussion

In order to apply the PSTP framework one needs to obtain the probability distributions of the uncontrollable events. These can reflect the beliefs of the domain encoder about their occurrence or can be estimated with experimentation. In addition, a system can learn and calibrate them by using the previous observations of the events.

Even though we presented a way to solve the static optimization schedule problem, there are still many limitations to our approach. First of all, the method does not necessarily provide the global optimal. We are investigating cases where the global optimal can be found, for example by the use of Lagrange multipliers or similar optimization techniques. Another (minor) limitation is the assumption that the conditional pdf are time-invariant. In the example of Figure 1(b) a better model would use a pdf for the event AO that depends on the time we leave home so that, for example, it would take longer on average to reach the office if we leave during rush hour. The use of assumption (1) for time invariance, was used in Equation (8) to derive the partial derivatives of $P(\text{Success}|T)$. If we know exactly how $p(x | Pa(x)=t)$ changes with t , then it is possible that we could still calculate the partial derivative and then assumption (1) could be dropped.

Solving the problem without assumption (2) holding is harder. Recall that assumption (2) was used in Equation (5) to derive $\log P(\text{Success}|T)$ and in order to split the probability of the conjunction of a number of constraints holding to the product of the probability of single constraints being satisfied. For example, $P(x-y \leq 5 \wedge x-TR \leq 10)$ when both x and y are random variables, does not necessarily equal $P(x-y \leq 5)P(x-TR \leq 10)$ because the satisfaction of the two constraints are not two independent events. Assumption (2) restricts us from being able to have PSTPs in which the synchronization of two uncontrollable events is required, e.g. two trucks arriving at around the same time. Finally, assumption (3) prohibits the use of pdf with discontinuities in their graph such as the uniform distribution. Nevertheless, such distributions could typically be adequately approximated by continuous ones.

A restriction of the PSTP model is the use of conditional pdf that depend only on the value of a single parent. The model is geared towards representing uncertain durations of actions in planning and scheduling for which a single parent is typically enough. Nevertheless, one can think of situations where this model is inadequate. For example, if the duration of heating up the engines of a spacecraft depends on when we started the action, but also on the amount of power we are willing to spend, then the probability distribution for the end of this action would require two parent variables: the time of the start of the heating action (temporal variable) and the amount of power used (non-temporal variable). It is possible to extend PSTPs to allow for multiple parent variables and optimally schedule events in the same fashion as for the single parent case presented in this paper. We are currently working to overcome these limitations and extend our model.

PSTPs can be derived from temporal planners in various ways. The planner can either use STPs or STPUs to generate a temporal plan that can be directly converted to a PSTP for execution. Alternatively, a planner might directly reason with PSTPs, perform a search in the space of PSTP plans and accept one that is above a threshold of probability of correct execution.

We would like to add that other related formalisms that deal with uncertainty are Markov Decision Processes (MDP) [9] and Influence Diagrams [10]. However, MDPs use discrete time while Influence Diagrams typically do not reason with continuous decision variables.

7. Conclusions

We presented a new constrained-based and probabilistic temporal reasoning formalism called the Probabilistic Simple Temporal Problem (PSTP) for representing plans and schedules with uncontrollable events and uncertain durations. We also provided methods that under certain conditions discover the schedule that maximizes the probability of executing the plan in a way that respects the temporal constraints. PSTP significantly extends previous formalisms that also deal with uncertainty of duration of actions by allowing the representation of and reasoning with probabilistic information. Unlike other models such as the STPU, uncertainty is represented in a principled and intuitive way and does not require guessing of the best approximate bounds. There are still restrictions in our model, such as the assumption that no constraints exist between two uncontrollable variables, and reasoning limitations, such as being incapable of analytically calculating the optimal dynamic execution strategy. We are currently researching ways to overcome the limitations.

References

1. Weld, D.S., *An Introduction to Least Commitment Planning*. AI Magazine, 1994. **15**(4): p. 27-61.
2. Jonsson, A., et al. *Planning in interplanetary space: theory and practice*. in *Artificial Intelligence Planning and Scheduling (AIPS-00)*. 2000.
3. Ghallab, M. and H.e. Laruelle, *Representation and Control in IxTeT, a Temporal Planner*, in *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems (AIPS-94)*. 1994. p. 61-67.
4. Dechter, R., I. Meiri, and J. Pearl, *Temporal constraint networks*. Artificial Intelligence, 1991. **49**: p. 61-95.
5. Vidal, T. and P. Morris. *Dynamic Control of Plans with Temporal Uncertainty*. in *IJCAI-2001 (to appear)*. 2001.
6. Chleq, N. *Efficient Algorithms for Networks of Quantitative Temporal Constraints*. in *Constraints'95*. 1995.
7. Vidal, T. and H. Frasier, *Handling consistency in temporal constraint networks: from consistency to controllabilities*. Journal of Experimental and Theoretical Artificial Intelligence, 1999. **11**: p. 23-45.
8. Hock, W. and K. Schittowski, *A comparative performance evaluation of 27 nonlinear programming codes*. Computing, 1983. **30**: p. 335.
9. White, D.J., *Markov Decision Processes*. 1993: John Wiley and Sons.
10. Zhang, N.L. *Probabilistic Inference in Influence Diagrams*. in *Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*. 1998.