A Column Generation Approach to Correlated Simple Temporal Networks

Andrew Murray[1], **Ashwin Arulselvan**[1], **Michael Cashmore**[1], **Marc Roper**[1], **Jeremy Frank**[2]

[1] University of Strathclyde, Glasgow, UK
[2] NASA AMES Research Center, Moffett Field, CA
{a.murray, ashwin.arulselvan, michael.cashmore, marc.roper}@strath.ac.uk, jeremy.d.frank@nasa.gov

Abstract

Probabilistic Simple Temporal Networks (PSTN) represent scheduling problems under temporal uncertainty. Strong controllability (SC) of PSTNs involves finding a schedule to a PSTN that maximises the probability that all constraints are satisfied (robustness). Previous approaches to this problem assume independence of probabilistic durations, and approximate the risk by bounding it above using Boole's inequality. This gives no guarantee of finding the schedule optimising robustness, and fails to consider correlations between probabilistic durations that frequently arise in practical applications. In this paper, we formally define the Correlated Simple Temporal Network (Corr-STN) which generalises the PSTN by removing the restriction of independence. We show that the problem of Corr-STN SC is convex for a large class of multivariate (log-concave) distributions. We then introduce an algorithm capable of finding optimal SC schedules to Corr-STNs, using the column generation method. Finally, we validate our approach on a number of Corr-STNs and find that our method offers more robust solutions when compared with prior approaches.

1 Introduction

Simple Temporal Networks with Uncertainty (STNU) (Vidal and Ghallab 1996) are graphs used to represent and reason over scheduling problems involving uncertain durations. A solution to an STNU is a schedule at which to execute a number of time-points, such that the temporal constraints are satisfied. STNUs capture uncertainty in the problem through the inclusion of set-bounded *contingent links*, over which the operator has no control. *Continuous probability distributions* are a more accurate representation of duration uncertainty; Probabilistic Simple Temporal Networks (PSTNs) model the uncertain duration with a probability density function (Tsamardinos 2002; Fang, Yu, and Williams 2014).

When dealing with uncertainty in temporal networks, it is typical to classify the problem in terms of controllability (Vidal 1999), which states how sophisticated an execution strategy is allowed. Strong Controllability (SC) asks if there is a single schedule robust to all uncertain outcomes, i.e. all constraints are satisfied no matter what happens. However, PSTNs are rarely strongly controllable as the continuous probability distributions are unbounded.

A PSTN can be reduced to a strongly controllable STNU through *truncating* the distributions over durations. However, this discards some of the probability mass of the distribution, thus reducing the robustness of the schedule. A variety of approaches have been introduced for solving SC of PSTNs while maximising robustness (Tsamardinos 2002; Fang, Yu, and Williams 2014; Santana et al. 2016; Lund et al. 2017). However, these approaches either bound above the risk using Boole's inequality which offers no guarantee on optimising robustness; or solve a generic non-linear optimisation problem which can be computationally expensive. Furthermore, all prior approaches assume independence of uncontrollable outcomes, which does not always hold. As an example consider a network of drones to be used in the delivery of medical supplies to rural communities (Filippi et al. 2022). The drone must fly between locations, picking up and dropping off supplies before they expire. Each leg of the journey is subject to correlated temporal uncertainty as a result of weather factors such as wind speed and direction. In vehicle routing problems correlation has been shown to exist in travel times (Bakach et al. 2021), with coefficients as high as 0.75 (Park and Rilett 1999).

In this paper we introduce the Correlated Simple Temporal Network (Corr-STN) which generalises the PSTN by removing the assumption of independence. We show that the problem of optimising robustness is convex for a wide range of log-concave distributions. This allows us to solve Corr-STN SC using one of the many available convex optimisation algorithms. We introduce one such approach leveraging the column generation method (Gondzio, González-Brevis, and Munari 2016), in which we iteratively refine and optimise on an approximation of the distributions. This approach provides the decision maker the choice to trade-off numerical time spent with an acceptable optimality guarantee.

We test our approach on a number of drone delivery temporal planning problems and compare results against a linear program (LP) using Boole's inequality (Santana et al. 2016) and our approach assuming independence. We then perform Monte-Carlo simulations, simulating the execution of each schedule on the Corr-STN and compare the robustness (the total probability of success) and accuracy of solutions. We show that considering the correlations offers more robust schedules than using Boole's or assuming independence. Although the robustness benefit of considering correlations varied substantially, we typically experienced greater improvements when the optimal robustness was low. When the optimal robustness was less than 0.5, considering correlation offered an average robustness improvement of 8.51% over using Boole's, and 3.50% over assuming independence. We also highlight that while Boole's is a bounding approximation of the true robustness, it can be grossly inaccurate, whereas assuming independence can be more accurate but is not guaranteed to give a conservative estimate of robustness. On the other hand, considering the correlation gives an accurate approximation of the true robustness but is more computationally expensive versus the other two approaches.

In Section 2 we introduce the definitions for PSTN SC. In Section 3 we place the contribution of this paper in context with respect to related work. In Section 4 we motivate the importance of considering correlations through reference to a toy example and formally define the Corr-STN. In Section 5 we highlight how Corr-STN SC can be encoded as a convex optimisation problem. In Section 6 we present one possible solution approach utilising the column generation method. In Section 7 we describe the setup and results of our experimental evaluation. We conclude and address avenues for future research in Section 8.

2 Background

A Simple Temporal Network with Uncertainty (STNU) (Vidal and Ghallab 1996) is a graph in which the nodes correspond to *time-points* and the edges (*links*) correspond to durations between the time-points. In STNU semantics, a distinction is made between *contingent links*, for which the duration of the interval is uncertain, and *requirement links* for which we can choose the duration.

Definition 1 (STNU)**.** *A STNU is a tuple, $S^U = \langle T_c, T_u, C, G\rangle$ where $b_i \in T_c$ is the set of controllable time-points and $e_i \in T_u$ is the set of un-controllable time-points, such that $t_i \in \{T_c \cup T_u\}$. The set C is the set of temporal requirement constraints between two time-points, normally written in the form $c(t_j, t_i) = t_j - t_i \in [l_{c,ij}, u_{c,ij}]$. The set G is the set of contingent links given in the form $g(e_i, b_i) = e_i - b_i \in [l_{g,i}, u_{g,i}]$. Here $l_{c\vee g}, u_{c\vee g}$ is the upper and lower limits for the constraint or contingent link respectively. Let $s(b) \in \mathcal{R}^+$ be the assignment of a value to the controllable time-point b. Let $o(e) \in \mathcal{R}^+$ be the value observed by an uncontrollable time-point e. A projection of a contingent link g_i is $\omega_i := o(e_i) - s(b_i)$.*

The key challenge is that the set of contingent links may take any random value within their bounds, and therefore an effective execution strategy must consider all possible projections for each contingent link.

Strong Controllability Controllability of an STNU can be separated into 3 categories (strong, dynamic, weak) and can be considered as a way of classifying how much control is needed to satisfy all constraints (Vidal 1999).

Definition 2 (Strong Controllability)**.** *Denote Ω, the space of outcomes of the contingent links: $\Omega = \times_{g\in G}[l_g, u_g]$. Let the schedule s be the assignment: $s(b), \forall b \in T_c$. An STNU S is said to be strongly controllable if: $\exists s \mid \forall\omega \in \Omega$, s satisfies all constraints.*

SC is a highly desirable property, as it offers the advantage that a fixed-time schedule can be computed offline which will work regardless of how the contingent links are realised at execution.

Without loss of generality, the requirement constraints can be separated into the set of controllable constraints C_c in the form: $c(b_j, b_i) = b_j - b_i \in [l_{c,ij}, u_{c,ij}]$ and uncontrollable constraints C_u in the form: $c(e_j, b_i) = e_j - b_i \in [l_{c,ij}, u_{c,ij}]$ or $c(b_j, e_i) = b_j - e_i \in [l_{c,ij}, u_{c,ij}]$, such that we can write uncontrollable constraints as:

$$c(e_j, b_i) = b_j + \omega_j - b_i \in [l_{c,ij}, u_{c,ij}] \tag{1}$$
$$c(b_j, e_i) = b_j - b_i - \omega_i \in [l_{c,ij}, u_{c,ij}] \tag{2}$$

To check whether an STNU S is strongly controllable, it is sufficient to check that the requirement constraints are satisfied for the worst possible projection of the contingent links. For constraints of the form (1) we have:

$$\max_{\omega_j\in[l_{g,j},u_{g,j}]} c(e_j, b_i) := u_{g,j} \leq b_i - b_j + u_{c,ij} \tag{3}$$
$$\min_{\omega_j\in[l_{g,j},u_{g,j}]} c(e_j, b_i) := l_{g,j} \geq b_i - b_j + l_{c,ij} \tag{4}$$

And for constraints of the form (2):

$$\max_{\omega_i\in[l_{g,i},u_{g,i}]} c(b_j, e_i) := u_{g,i} \leq b_j - b_i - l_{c,ij} \tag{5}$$
$$\min_{\omega_i\in[l_{g,i},u_{g,i}]} c(b_j, e_i) := l_{g,i} \geq b_j - b_i - u_{c,ij} \tag{6}$$

For further details on solving STNU SC, we refer the reader to a relevant paper (Morris and Muscettola 2005; Cimatti, Micheli, and Roveri 2015).

Probabilistic Simple Temporal Networks When sufficient data is available, it is more accurate to model the space of projections of a contingent link by a probability density function. This allows the scheduling process to focus on the durations *most likely* to be realised at execution. Probabilistic Simple Temporal Networks (PSTN) were introduced in (Tsamardinos 2002; Fang, Yu, and Williams 2014).

Definition 3 (PSTN)**.** *A PSTN is a tuple, $S^P = \langle T_c, T_u, C, D\rangle$, where T_c, T_u and C are as per the STNU. The set of probabilistic constraints D, are in the form $d(e_i, b_i) = e_i - b_i = X_i$, where X_i is a random variable with a set of outcomes Ω_i, probability density function $f(\omega_i)$ and cumulative probability function $F(z) = P(X_i \leq z)$.*

PSTN SC and Risk in Literature It is impossible to find a schedule that will work for all values in an unbounded distribution. As a result, it is typical to truncate the distribution by neglecting the extreme, unlikely outcomes in the tails, i.e: $\Omega_{*i} = [l_{d,i}, u_{d,i}]$. If we denote by $d_*(e_i, b_i)$ the transformed probabilistic constraint with value defined by random variable X_{*i} and set of outcomes Ω_{*i}, then performing this transformation transforms the probabilistic constraint to a contingent link, i.e. $d_*(e_i, b_i) = e_i - b_i = X_{*i} \in [l_{d,i}, u_{d,i}] \equiv g(e_i, b_i)$. Applying this transformation to all $d \in D$ is equivalent to transforming the PSTN, S^P to an equivalent STNU, S^{*U}. However the schedule is now only robust to the outcomes considered in S^{*U} (see Figure 1).

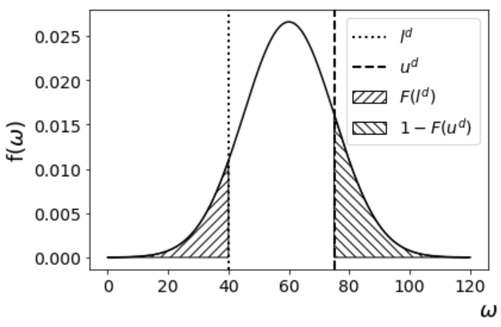


Figure 1: Figure showing risk associated with squeezing a probabilistic constraint to an equivalent contingent link.

The probability mass excluded by performing this transformation is the *risk* of S^{*U}, while the probability mass considered is the *robustness*. We refer to Fang, Yu and Williams (2014) for a definition of robustness and risk, written in its equivalent form below.

Definition 4 (Robustness and Risk)**.** *We denote $\Omega_R \subseteq \Omega$ and $c(\omega)$ the value of each constraint $c \in C$ given an outcome ω. If $\omega \in \Omega$ and for every $c \in C$, $c(\omega) \in [l_{c,ij}, u_{c,ij}]$: then $\omega \in \Omega_R$. The robustness Γ, is $P(\Omega_R)$, while the risk Δ, is $P(\bar{\Omega}_R)$, where $\bar{\Omega}_R$ denotes the complement of the set Ω_R.*

Since the joint probability functions, $P(\Omega_R)$ and $P(\bar{\Omega}_R)$ may be non-trivial, it is typical to treat each probabilistic constraint independently, such that the robustness can be evaluated: $\Gamma = \prod_{d\in D} P(l_d \leq X \leq u_d)$ and the risk: $\Delta = 1 - \prod_{d\in D}(P(l_d \leq X \leq u_d))$. The values of u_d and l_d are determined through the SC relationships, equations (3) to (6), by substituting l_g, u_g for l_d, u_d. To permit a linear formulation, it is possible to bound above the risk using Boole's inequality. The robustness can then be approximated through: $\Gamma = \sum_{d\in D}(F(u_d) - F(l_d))$, while the risk: $\Delta = \sum_{d\in D}(1 - F(u_d) + F(l_d))$.

3 Related Work

Tsamardinos (2002) takes a risk minimisation approach to PSTN SC, and makes use of assumptions to leverage Sequential Quadratic Programming. Santana et al. (2016) and Lund et al. (2017) make varying assumptions to permit the use of LPs to allocate risk in PSTNs. Santana et al. pose and solve an LP using Boole's inequality and piecewise linear approximations of the cumulative density function (CDF); Lund et al. iteratively pose and solve an LP using non-bounding approximations of the probability mass on the tails of the probability distributions.

Fang et al. (2014) introduced the CC-PSTN: by enforcing an allowable tolerance on the risk as a constraint in the system. Some other objective function could then be optimised, while ensuring that the schedule risk does not exceed the bound. In some instances, the risk required to enforce SC can be deemed too high. Yu et al. (2015) extend the chance-constrained framework to the *Relaxable* CC-PSTN by permitting the use of soft constraints which can be relaxed. The relaxable CC-PSTN is solved by Yu et al. using a nonlinear solver, combined with a conflict detection mechanism based on identification of negative cycles in STNUs.

To the best of the authors' knowledge, all previous approaches to PSTN SC assume independence (Lund et al. 2017; Fang, Yu, and Williams 2014; Yu, Fang, and Williams 2015; Santana et al. 2016; Tsamardinos 2002), and either use Boole's inequality to bound above the risk (Fang, Yu, and Williams 2014; Yu, Fang, and Williams 2015; Santana et al. 2016), or solve a generic non-linear optimisation problem (Tsamardinos 2002; Wang and Williams 2015). Using Boole's inequality permits the use of LPs, however it is not guaranteed to return the optimal solution maximising robustness (see Section 4); procedures used to solve generic non-linear optimisation problems offer no guarantee on either optimality or computational efficiency.

The main contributions of this paper is that we show in Section 5 that PSTN SC can be modelled as a convex optimisation problem enabling globally optimal, robust schedules to be found. We suggest one such approach to solve this problem in Section 6, that has been employed in convex optimisation literature. Rather than using piecewise linear approximations of the CDF which is not convex (see Santana (2016)), we form inner approximations of the negative log of the CDF which is convex. These inner approximations are analogous to piecewise linear approximations in one dimension, however generalise to polyhedral approximations at higher dimensions. This makes it possible to approximate multivariate random variables and consequently consider correlations in the optimisation, which we motivate in the coming section.

4 Motivating Example

We motivate our approach by discussing the toy example given in Figure 2. Consider a drone used in the transportation of medical supplies. After being notified of a potential delivery, the drone must fly from a depot to a location to pick it up. The travel time of this leg of the journey $(e_1 - b_1)$ is described by the random variable $X_1 \sim \mathcal{N}(60, 10)$. After it has collected the supplies it must fly to the drop off point and deliver the supplies within the required time-frame (between 0 and 160 minutes after setting out $e_2 - b_1$). This leg of the journey $(e_2 - b_2)$ can also be described by the random variable $X_2 \sim \mathcal{N}(100, 25)$. We want to find the schedule that maximises robustness Γ. We have two uncontrollable constraints $c(b_2, e_1)$ and $c(e_2, b_1)$. From (1) and (2) we have:

$$c(b_2, e_1) \equiv 0 \leq b_2 - b_1 - X_1 \leq \infty$$
$$c(e_2, b_1) \equiv 0 \leq b_2 + X_2 - b_1 \leq 160$$

The only decision variable is the difference between the time assigned to b_2 and b_1. W.l.o.g. we assume $b_1 = 0$.

Boole's Inequality Using Boole's inequality we can formulate the objective as:

$$\max_{b_2}\{P(b_2 - \infty \leq X_1 \leq b_2) + P(-b_2 \leq X_2 \leq -b_2 + 160)\}$$

If we consider first that $b_2 = 75$ then we have:

$$(F_{X_1}(75) - F_{X_1}(-\infty)) + (F_{X_2}(85) - F_{X_2}(-75)) = 1.21$$

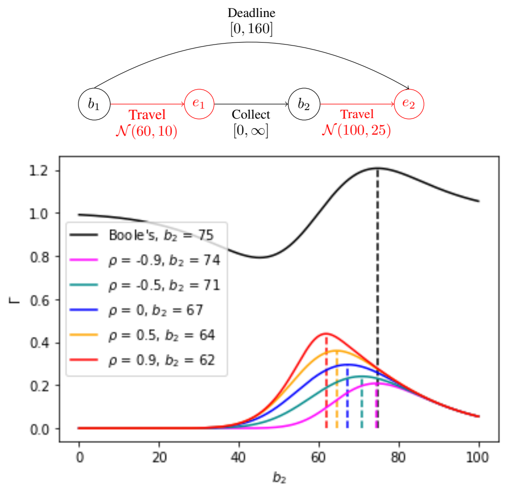


Figure 2: Image showing toy example (above) and comparison of robustness using Boole's versus actual robustness with varying correlation coefficient ρ (below).

Next we consider that $b_2 = 67$ so we have:

$$(F_{X_1}(67) - F_{X_1}(-\infty)) + (F_{X_2}(93) - F_{X_2}(-67)) = 1.14$$

Using Boole's, b_2 = 75 is clearly the better schedule.

Joint Outcome with Independence If we consider the joint outcome with independence then the objective is:

$$\max_{b_2}\{P(b_2 - \infty \leq X_1 \leq b_2)P(-b_2 \leq X_2 \leq -b_2 + 160)\}$$

For $b_2 = 75$:

$$(F_{X_1}(75) - F_{X_1}(-\infty))\,(F_{X_2}(85) - F_{X_2}(-75)) = 0.26$$

And $b_2 = 67$:

$$(F_{X_1}(67) - F_{X_1}(-\infty))\,(F_{X_2}(93) - F_{X_2}(-67)) = 0.30$$

Considering the joint outcome, the optimal solutions are switched with the schedule $b_2 = 67$ offering a 15% increase in robustness versus the solution returned using Boole's inequality. This effect is observed in greater detail in Figure 2.

Joint Outcome with Correlation We will now show that, even considering the joint outcome with independence is not necessarily guaranteed to return optimal solutions if the correlation is experienced when the schedule is executed.

We return to the drone example and consider that the travel times X_1 and X_2 are correlated due to uncertainty in wind speed. We plotted the robustness for varying b_2 and varying correlation coefficient ρ in Figure 2. Considering a fixed schedule of $b_2 = 67$, with $\rho = 0$ we have independence and consequently the robustness is as per the previous section $\Gamma = 0.30$. On the other hand, if the two variables have correlation $\rho = 0.9$ then the robustness $\Gamma = 0.39$. If we were to assume independence in the optimisation then this is the best robustness we can hope for. It is clear from Figure 2, that better robustness can be achieved through scheduling b_2 five minutes earlier ($b_2 = 62$) such that the robustness $\Gamma = 0.44$. Considering correlation in the scheduling process offers a 12.8% improvement in robustness.

To explain this difference we refer to Figure 3, which shows a contour plot of the joint probability density function of $X = [X_1, X_2]$. With $b_2 = 67$, the robustness is given by the volume beneath the contour plot within a rectangle with dimensions $-\infty \leq X_1 \leq 67$ and $-67 \leq X_2 \leq 93$ (shown by the vertical and horizontal blue lines). The dimensions of the box are fixed by the constant bounds, $[l^c, u^c]$ associated with each uncontrollable constraint. Finding the schedule that optimises robustness, involves moving the box (by changing the schedule), such that it covers as much of the probability mass as possible. This in turn is dependent on the underlying distribution - for which correlation may have a significant effect. For the correlated case the optimal occurs at $b_2 = 62$, such that the rectangle has bounds $-\infty \leq X_1 \leq 62$ and $-62 \leq X_2 \leq 98$ (shown by the vertical and horizontal red lines).

Objective Accuracy It's worth noting that if you assume independence, you are not guaranteed to have a conservative estimate of the actual robustness. For example in Figure 2, with $\rho = 0$, the robustness from the model would be $\Gamma = 0.3$. The decision maker would be expected to make a decision based on this value, whereas in reality the robustness experienced could be much lower. If correlation $\rho = -0.9$ was experienced at execution time, the actual robustness from the schedule $b_2 = 67$ would be $\Gamma = 0.16$. Referring to Figure 3 can offer some insight into this effect. When we solve assuming independence, the solution is a conservative approximation of the probability mass under the blue contour plot, enclosed within the blue box. The actual robustness is the probability mass under the red contour plot (also within the blue box). This is not guaranteed to be strictly less than the equivalent probability mass under the independent pdf. While the optimal schedule for Boole's and correlation $\rho = -0.9$ are similar, the objective of Boole's, $\Gamma = 1.2$ offers nothing to a decision maker who has to reason over the risk of the schedule. Under such circumstances it becomes necessary to consider the correlation directly. We now formally introduce the Corr-STN:

Definition 5 (Corr-STN). *A Corr-STN is a tuple, $S^C = \langle T_c, T_u, C, D, R\rangle$, where T_c, T_u and C are as per the PSTN. R is the set of correlations involving a number of probabilistic constraints with correlation matrix ϱ. Each $r \in R$ defines an n dimensional multivariate normal vector $X \sim (\mu, \Sigma)$ with mean vector μ and covariance matrix Σ. The set D is the set of independent probabilistic constraints. If there are no correlations, then the set $R = \varnothing$ and the set D is the set of all independent probabilistic constraints as per the PSTN.*

5 Corr-STN SC is Convex

In this section we show how Corr-STN SC can be formulated as a convex optimisation problem.

Decision Variables The decision variables include the vector of controllable time-points x, and the vector of lower and upper bounds z (introduced below).

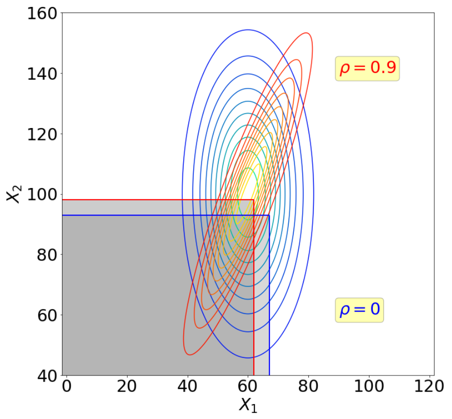


Figure 3: Image showing bi-variate normal distribution probability density function with and without correlation.

Controllable Constraints The controllable constraints can be written in the form of two less than inequalities: $b_j - b_i \leq u_{c,ij}$ and $b_i - b_j \leq -l_{c,ij}$, such that they represent a polyhedron: $Ax \leq \beta$, where A is the coefficient matrix of values, $A_{ij} \in \{-1, 1, 0\}$, β is the vector of bounds $\beta_i \in \{u_c, -l_c\}$ and x is the decision variable vector.

Independent Probabilistic Constraints For each $d \in D$, we have a number of uncontrollable constraints preceding/succeeding it. From (3) to (6), we can write the uncontrollable constraints in the form: $b_i - b_j + l_{c,ij} \leq X_j \leq b_i - b_j + u_{c,ij}$ and $b_j - b_i - u_{c,ij} \leq X_i \leq b_j - b_i - l_{c,ij}$. Consequently we have the matrix inequality $z_d \leq T_d x + q_d$, where $z_{d,i} \in \{u_d, -l_d\}$, $T_{d,ij} \in \{-1, 1, 0\}$ and $q_{d,i} \in \{u_c, -l_c\}$. The probability that the constraint is satisfied is given by the probability function $F_d = P(l_d \leq X \leq u_d)$.

Correlations For each correlation $r \in R$, we write the n_r uncontrollable constraints involved in the correlation in the form: $z_r \leq T_r x + q_r$. The difference here is that we have more than one random variable involved in each correlation. The vector of upper and lower bounds are in the form $u_r = [u_{r,1}, \ldots, u_{r,n_r}]^T$ and $l_r = [l_{r,1}, \ldots, l_{r,n_r}]^T$, and therefore calculating the probability that the constraints are satisfied involves calculating the joint probability function $F_r = P(l_r \leq X \leq u_r)$, for the multivariate distribution $X \sim \mathcal{N}(\mu, \Sigma)$ as per Definition 5.

Objective Function For each $d \in D$ and $r \in R$, we know that the lower and upper bounds l_d, u_d and l_r, u_r are directly encoded in the vectors z_d and z_r which represent rows of a vector z, such that $z = [z_{d_1}, \ldots, z_{d_{|D|}}, z_{r_1}, \ldots, z_{r_{|R|}}]^T$. The objective function is to maximise the robustness Γ giving the following optimisation problem:

$$\max_{x,z}\{\prod_{r\in R} F_r \prod_{d\in D} F_d \;\mid\; z \leq Tx + q,\; Ax \leq \beta\}$$

On Convexity Following from above, the only non-linear function here is the robustness $\Gamma = \prod_{r\in R} F_r \prod_{d \in D} F_d$, used in the objective. If the probability distribution of a random vector X is log-concave, then the cumulative probability function is also log-concave and so is any linear transformation of it (Prékopa 1971, 1973). The result is that the functions F_r and F_d are log-concave. It should be noted that many interesting distributions contain this characteristic (Prékopa 2003), the multivariate normal distribution being one such family. Another important property of log-concave functions is that the product of log-concave functions is also log-concave, and therefore:

$$\log\left(\prod_{r\in R} F_r \prod_{d\in D} F_d\right) = \sum_{r\in R} \log F_r + \sum_{d\in D} \log F_d$$

is concave. As a result we can reformulate the optimisation problem as a convex one:

$$\min_{x,z}\{\sum_{r\in R} \phi_r + \sum_{d\in D} \phi_d \;\mid\; z \leq Tx + q,\; Ax \leq \beta\} \tag{7}$$

Where $\phi_r = -\log F_r$ and $\phi_d = -\log F_d$.

Running Example Consider the small Corr-STN from Figure 2. In this example, we have no controllable constraints, no independent probabilistic constraints and only one correlation defining a multivariate normal distribution $X = [X_1, X_2]$. The uncontrollable constraints associated with the correlation are $c(b_2, e_1)$ for X_1 and $c(e_2, b_1)$ for X_2 respectively. We encode this as:

$$\begin{bmatrix} u_{r,1} \\ u_{r,2} \\ -l_{r,1} \\ -l_{r,2} \end{bmatrix} \leq \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 160 \\ \infty \\ 0 \end{bmatrix}$$

The objective is to find the value of vectors $x = [b_1, b_2]^T$, and $z = [u_{r,1}, u_{r,2}, -l_{r,1}, -l_{r,2}]^T$ that minimises ϕ_r.

6 Method

The key result of the previous section is that the problem is convex, allowing use of a rich suite of existing solution methods. For a recent survey, we refer to Van Ackooij (2020). In the coming section, we introduce one such method that forms an inner approximation of the convex functions ϕ_d and ϕ_r using a number of generated approximation points (hereby referred to as columns) (Fábián et al. 2018; Fábián 2021). The problem is then solved via the column generation procedure as outlined in Algorithm 1, in which two optimisation phases are iteratively solved:

1. The **Restricted Master Problem** (RMP), which solves the probability maximisation problem using the columns generated so far (line 4).
2. A **Column Generation Problem** (CGP) for each function ϕ_d and ϕ_r, which finds the best new column to include in the RMP (line 7).

The results of the RMP and CGP are stored in modelR and modelC respectively. We extract the dual values (modelR.duals) from the solution to the RMP, and use them to model the *reduced cost* which we set as the objective to the CGP (modelC.objective). Since we are minimising reduced cost, any column whose reduced cost is negative is called an *improving* column. If an improving column can be found (line 8), we set terminate to False (line 9), and add the new column (line 10). The process then repeats until no improving column can be found.

Algorithm 1: Algorithm for SC of Corr-STN

```
    Input  : A Corr-STN, $S^C$
    Output: An optimisation model $modelR$ containing
            a schedule that optimises robustness.
 1  $columns \leftarrow$ getInitialColumn($S^C$);
 2  $terminate \leftarrow$ False;
 3  while $terminate \leftarrow False$ do
 4      $modelR \leftarrow$ RMP($columns, S^C$);
 5      $terminate \leftarrow$ True;
 6      for $function \in D \cup R$ do
 7          $modelC \leftarrow$ CGP($modelR$.duals, $function$);
 8          if $modelC$.objective $< 0$ then
 9              $terminate \leftarrow$ False;
10              $columns$.add($modelC$.solution)
11          end
12      end
13  end
    Return: $modelR$
```

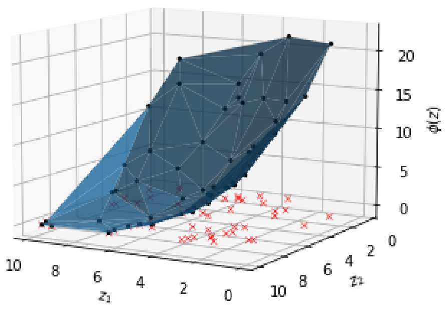


Figure 4: Inner approximation for a bivariate convex function $\phi(z)$. The red crosses are the approximation points (columns) z^i at which the function has been evaluated and the black dots are the function evaluations ϕ^i.

Inner Approximation Note that ϕ_r and ϕ_d are the only nonlinear functions in (7) and they are convex. For any convex function ϕ, if we have enumerated sufficiently many finite points, $\{z^1, z^2, ..., z^K\}$, referred to as *base* (see (Geoffrion 1970)), in its domain, and let $\phi^i := \phi(z^i)$, then we can approximate $\min \phi(z)$ with :

$$\min\{\sum_{i=1}^{K} \phi^i \lambda^i : \sum_{i=1}^{K} \lambda^i = 1, \lambda^i \geq 0\} \tag{8}$$

where K depends on the desired level of approximation (see Figure 4). Note that from henceforth the notation ^{i} refers to the i^{th} column. Since the inner approximation is an over-estimate of the convex functions ϕ_r and ϕ_d, it is a conservative approximation of the robustness F_r and F_d.

Running Example To highlight this we return to the running example. Assume that we have evaluated the function ϕ_r at a number of points: l^i, u^i for $i = 1, 2, .., K$, such that

$$\min_{x,\lambda} \sum_{r \in R} \sum_{i=1}^{K_r} \phi_r^i \lambda_r^i + \sum_{d \in D} \sum_{i=1}^{K_d} \phi_d^i \lambda_d^i$$

$$\begin{aligned}
s.t.\ \ & Ax \leq \beta \\
& \sum_{i=1}^{K_r} \lambda_r^i z_r^i \leq T_r x + q_r && r \in R\ (dual : \pi_r) \\
& \sum_{i=1}^{K_d} \lambda_d^i z_d^i \leq T_d x + q_d && d \in D\ (dual : \pi_d) \\
& \sum_{i=1}^{K_r} \lambda_r^i = 1 && r \in R\ (dual : \nu_r) \\
& \sum_{i=1}^{K_d} \lambda_d^i = 1 && d \in D\ (dual : \nu_d) \\
& x_i,\ \lambda^i \geq 1
\end{aligned}$$

Figure 5: Master Problem

ϕ_r^i refers to $\phi_r(l^i, u^i)$. The inner approximation would be:

$$\begin{bmatrix} u_{r,1}^1 & \cdots & u_{r,1}^K \\ u_{r,2}^1 & \cdots & u_{r,2}^K \\ -l_{r,1}^1 & \cdots & -l_{r,1}^K \\ -l_{r,2}^1 & \cdots & -l_{r,2}^K \end{bmatrix} \begin{bmatrix} \lambda_r^1 \\ \vdots \\ \lambda_r^K \end{bmatrix} \leq \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 160 \\ \infty \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \ldots 1 \end{bmatrix} \begin{bmatrix} \lambda_r^1 \\ \vdots \\ \lambda_r^K \end{bmatrix} = 1, \quad \lambda_r^i \geq 0, \quad \phi_r = \sum_{i=1}^{K} \phi_r^i \lambda_r^i \tag{9}$$

While the column: $z^i = [u_{r,1}^i, u_{r,2}^i, -l_{r,1}^i, -l_{r,2}^i]$.

Restricted Master Problem If we form an inner approximation for each independent probabilistic constraint and correlation using (8), we can re-write (7) in its approximate form as a linear program as shown in Figure 5. We refer to this as the **Master Problem** (MP). Notice that we have replaced the z variables in (7) with λ variables. With each point, z^i, in our base we can associate a *column* of coefficients in the constraint matrix corresponding to the variable λ^i (as shown in (9)). The value of the lambda variables allow for the convex combination of the columns enumerated so far. We refer K_r, K_d as the number of columns generated for each correlation r and independent probabilistic constraint d. Likewise we refer λ_r as the K_r dimensional vector of variables associated with the columns of a correlation r and λ_d as the K_d dimensional vector of variables associated with the columns generated for an independent probabilistic constraint d.

K can be prohibitively large when solving the MP in Figure 5 directly. In addition, we are only interested in the columns from our base that are in the vicinity of the optimal solution. The key idea is that we iteratively solve a restricted version, that we refer to as the **Restricted Master Problem** (RMP), where only a subset, $\{z^1, \ldots, z^k\}$ such that $k << K$, of our base is considered. Note that the optimal solution to the RMP is always feasible to the MP. For it to be optimal, none of the unconsidered columns in our base would improve the objective when included in the RMP. If no such column exists, then the optimal solution of the RMP is also optimal to the MP.

Finding an Initial Feasible Point In order to initialise the algorithm, we must find an initial column z^0, for which the RMP has a feasible solution. Any previous PSTN SC algorithm can be used to generate a feasible point. To see this, consider that we obtain a schedule, i.e. an assignment of a value to all the controllable time-points: $x^0 \in \mathcal{R}^n$, which satisfies all the constraints. The equivalent column can then be evaluated as $z^0 = Tx^0 + q$. Many efficient PSTN SC algorithms exist capable of finding such a feasible point, for details on how to implement such an algorithm, we refer the reader to the relevant paper (Tsamardinos 2002; Fang, Yu, and Williams 2014; Santana et al. 2016; Lund et al. 2017).

Column Generation A column is said to be an *improving* column if the reduced cost is negative (Dantzig 1963). Given a linear program $\min_x\{c^T x \mid Ax \leq b,\ x_i \geq 0\}$, any variable x_i that takes a zero value in the simplex procedure is known as a *non-basic variable*. The *reduced cost* of introducing a non-basic variable x_i into the simplex basis is: $c_i - {A^i}^T y$, where A^i refers to column i of matrix A and y is the dual vector associated with the constraint $Ax \leq b$.

Running Example Returning to the ongoing example, we show how to generate an improving column z_r^{k+1}, for ϕ_r. The objective coefficient would be $c_{k+1} = \phi_r^{k+1}$. From (9), the column of coefficients in the constraint matrix associated with variable λ^{k+1} is $A^{k+1} = [z_r^{k+1}, 1]^T$ and from Figure 5 the dual vector $y = [\pi_r, \nu_r]$. We can therefore find the best improving column by minimising reduced cost, i.e. solving the following optimisation problem.

$$\min_{z_r}\{\phi_r(z_r) - z_r^T \pi_r - \nu_r\} := \min_{l_r, u_r}\{-\log F(l_r, u_r) \\ -[u_r, -l_r]\begin{bmatrix}\pi_{u_r}\\ \pi_{l_r}\end{bmatrix} - \nu_r \mid u_r > l_r\} \qquad (10)$$

Such that $\pi_{u_{r,i}} = \sum_{\{j: z_{r,j} = u_{r,i}\}} \pi_{r,j}$ refers to element i in vector π_{u_r}, where $j = 1, 2, ..., m$ and m is the number of rows in constraint matrix $z_r \leq T_r x + q_r$. Similarly $\pi_{l_{r,i}} = \sum_{\{j: z_{r,j} = -l_{r,i}\}} \pi_j$ refers to element i in vector π_{l_r}. To prevent domain errors with $\log(0)$, we constrain the upper bound to be greater than the lower bound, $u_r > l_r$.

We refer to this as the **Column Generation Problem** (CGP). We solve one CGP for all $r \in R$ and $d \in D$. If no improving column can be found for any function, at any iteration, then it is not possible to find another variable λ_{k+1} (and column z^{k+1}) which will improve the objective when entered into the simplex basis. In other words, our inner approximation already contains the optimal solution and so we can terminate the algorithm. The convergence of this procedure has been shown in (Dantzig 1963).

In order to solve (10) it is necessary to efficiently compute the gradient vector. This can be evaluated as:

$$\nabla\left(\phi(z_r) - z_r^T \pi_r - \nu_r\right) = -\frac{\nabla F(l_r, u_r)}{F(l_r, u_r)} - \begin{bmatrix}\pi_{u_r}\\ \pi_{l_r}\end{bmatrix}$$

There exists efficient algorithms capable of calculating cumulative probabilities of multivariate normal distributions (e.g. (Genz 1992)). Prekopa (2013) proves it is possible to analytically evaluate the gradient of the function $F(z)$ for multi-variate normal distributions using the same efficient algorithm (Van Ackooij et al. 2010, 2011). For a formula for the case, $\nabla F(l, u)$ which relies on the same result, we refer the reader to (Van Ackooij et al. 2010).

Stopping Criteria Using the objectives of the RMP and CGP's, we can derive bounds on optimality and terminate the algorithm when this bound falls below a certain threshold. We denote $\mathcal{M}$, the optimal objective to the RMP on iteration k, is a valid upper bound to the master problem. If $\mathcal{C}_r$ and $\mathcal{C}_d$ are the optimal solutions to the CGP associated with correlation r and independent probabilistic constraint d respectively on iteration k, then $\mathcal{M} + \sum_{r\in R}\mathcal{C}_r + \sum_{d\in D}\mathcal{C}_d$ is a valid lower bound on the optimal solution. We define some allowable tolerance, ε. After each iteration, we check the following condition: $(UB - LB)/LB \leq \varepsilon$, and terminate the algorithm when it is satisfied.

7 Experimental Evaluation

PDDL (Fox and Long 2003) problem instances were generated for a drone delivery domain. To generate Corr-STN instances, each problem was solved using the temporal planner OPTIC (Benton, Coles, and Coles 2012). The best plan found within 10 minutes was saved as an STN. For each STN, we then generate 10 separate PSTN instances by sampling mean and standard deviations to apply to actions. For each PSTN, we then create 3 separate Corr-STN instances by sampling random correlation matrices of size 2, 3 and 4. Finally TIL deadlines were then varied to generate Corr-STN problems with a wide variety of robustness. The result was a total of 5850 Corr-STN problem instances of which 4872 could be solved [1].

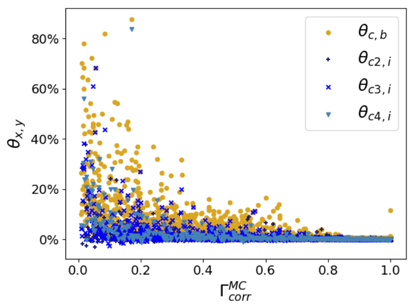
Each Corr-STN was then solved using three methods: an SC Linear Program with Boole's inequality (implementation of the PARIS algorithm from Santana et al. (2016)) (referred to as **Boole's**); Column Generation method assuming independence (referred to as **Independent**); Column Generation with correlation (referred to as **Correlated**). Python was used for the implementation with Gurobi as the linear programming optimiser, and the SLSQP solver within the python package SciPy as the column generation solver.

Robustness To assess the robustness we simulated each schedule, for each Corr-STN, 20,000 times and calculated the Monte-Carlo robustness. Boole's and independent robustness were then compared to correlated, and the percentage difference plotted in Figure 6a.

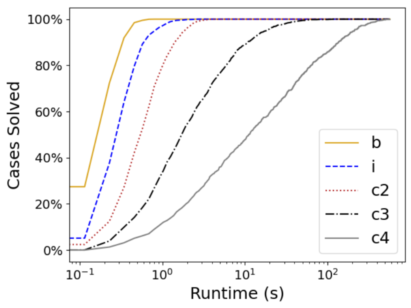
The improvement in robustness when we consider correlation versus using Boole's is substantial but has a wide variance. In general, we see a significant improvement on problems in which the optimal robustness is low with some cases offering up to 80% improvement. The intuition for this can be gained from Figure 3. As discussed in Section 4, the

[1] Source code and benchmark problems can be accessed online at: https://anonymous.4open.science/r/corr-stn/

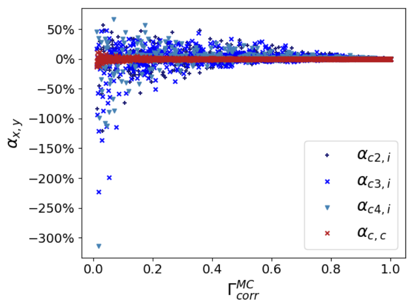
Note: Γ^{MC} and Γ^{TH} refer to the Monte Carlo robustness, and theoretical robustness obtained from the optimisation model. The notation b, i, c, $c2$, $c3$ and $c4$ refer to the results for Boole's, independent, all correlated cases, and correlated cases of sizes 2, 3 and 4 respectively. Figures 6a and 6c are plotted against the optimal probability, as obtained from the Monte Carlo simulations considering correlation.



(a) Plot showing % difference in Monte Carlo robustness for different solutions: $\theta_{x,y} = (\Gamma^{MC}_x - \Gamma^{MC}_y)/\Gamma^{MC}_x \times 100$. Boole's and independent are compared to correlated of different sizes.



(b) Plot showing % of cases solved versus runtime for Boole's, independent and correlated.



(c) Plot showing % difference in Monte Carlo and theoretical robustness: $\alpha_{x,y} = (\Gamma^{MC}_x - \Gamma^{TH}_y)/\Gamma^{MC}_x \times 100$. Theoretical for independent and correlated are compared to correlated Monte-Carlo.

Figure 6: Experimental results for (a) robustness, (b) runtime and (c) accuracy.

problem of Corr-STN SC involves moving a box (by varying the schedule) of n dimensions over the n dimensional multivariate normal pdf. When the box is small and constrained to the outer corners of the distribution, the optimal location of the box can be quite different. Of the 928 cases where the correlated Monte Carlo robustness was less than 0.5, correlated offered a mean improvement of 8.51% over Boole's and 3.50% over independent.

Runtime Figure 6b plots the percentage of cases solved versus runtime for the Boole's, independent and correlated of varying correlation sizes. As expected using the Boole's. LP is substantially faster with all cases solved within 1 second. This is a result of the fact that the encoding is entirely linear. On the other hand approximately 90% of the independent cases and 80% of the correlated size 2 cases could be solved within 1 second. The runtime grows exponentially with the size of the distribution, some cases with correlation size 4 took up to 400 seconds. Nonetheless approximately 80% of the correlated size 3 cases and 50% of the correlated size 4 cases could be solved within 10 seconds.

It's worth mentioning that the stopping criteria allows for a trade-off between the solution quality and runtime. All of the problems were solved with a gap of 1% (i.e. $\varepsilon = 0.01$), however it is possible to terminate the algorithm earlier, and return the best solution found so far.

Accuracy To measure accuracy, we compare the theoretical robustness obtained from the objective for independent and correlated, with the Monte-Carlo robustness considering the correlation. The percentage difference is plotted in Figure 6c. If we assume independence, we can obtain theoretical robustness values which are up to 3 times higher than the actual robustness observed through Monte-Carlo simulation. This is because assuming independence is not guaranteed to provide a bounding approximation of the correlated robustness (see Section 4 and Figure 3). We do not plot the theoretical versus experimental robustness using Boole's since the objective is not representative of the actual probability.

8 Conclusion

To summarise, we formally define the Corr-STN and show that Corr-STN SC is convex for a wide range of multivariate (log-concave) distributions. We introduce one solution method using column generation, in which we iteratively refine and optimise on an approximation of the distributions.

In our experimental validation we solved a number of Corr-STNs using 1. Boole's inequality, 2. column generation with independence and 3. column generation with correlations of varying sizes. We compared schedules in terms of robustness, runtime and accuracy. We find that for problems in which the optimal probability is small, considering correlation can offer a significant robustness improvement versus Boole's inequality and column generation with independence. Additionally, to ensure an accurate, bounding approximation of robustness, considering correlation is necessary, however it comes with additional computational expense which may be prohibitive for many applications.

While we have empirically shown that considering the correlation in the scheduling process can be important and have outlined a method for doing so, we note that the value of considering the correlation varies significantly. For some cases, the improvement is substantial, however for others it is not worth the additional computational effort. There are a vast number of problem specific factors which affect this: size/magnitude of correlation, tightness of constraints as well as which constraints we consider correlated to name a few. In future work we hope to define a metric which can be used to determine the benefit of considering the correlation for particular networks.

References

Bakach, I.; Campbell, A. M.; Ehmke, J. F.; and Urban, T. L. 2021. Solving vehicle routing problems with stochastic and correlated travel times and makespan objectives. *EURO Journal on Transportation and Logistics*, 10: 100029.

Benton, J.; Coles, A.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 22.

Cimatti, A.; Micheli, A.; and Roveri, M. 2015. Solving strong controllability of temporal problems with uncertainty using SMT. *Constraints*, 20: 1–29.

Dantzig, G. B. 1963. *Linear programming and extensions*. Princeton: Princeton University Press.

Fábián, C. I. 2021. Gaining traction: on the convergence of an inner approximation scheme for probability maximization. *Central European Journal of Operations Research*, 29(2): 491–519.

Fábián, C. I.; Csizmás, E.; Drenyovszki, R.; van Ackooij, W.; Vajnai, T.; Kovács, L.; and Szántai, T. 2018. Probability maximization by inner approximation. *Acta Polytechnica Hungarica*, 15(1): 105–125.

Fang, C.; Yu, P.; and Williams, B. 2014. Chance-constrained probabilistic simple temporal problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Filippi, G.; Vasile, M.; Patelli, E.; and Fossati, M. 2022. Generative optimisation of resilient drone logistic networks. In *Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC)*.

Fox, M.; and Long, D. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20: 61–124.

Genz, A. 1992. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2): 141–149.

Geoffrion, A. M. 1970. Elements of large-scale mathematical programming Part I: concepts. *Management Science*, 16(11): 652–675.

Gondzio, J.; González-Brevis, P.; and Munari, P. 2016. Large-scale optimization with the primal-dual column generation method. *Mathematical Programming Computation*, 8(1): 47–82.

Lund, K.; Dietrich, S.; Chow, S.; and Boerkoel, J. 2017. Robust execution of probabilistic temporal plans. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Morris, P. H.; and Muscettola, N. 2005. Temporal dynamic controllability revisited. In *Proceedings of the 20th National Conference on Artificial Intelligence*, volume 3.

Park, D.; and Rilett, L. R. 1999. Forecasting freeway link travel times with a multilayer feedforward neural network. *Computer-Aided Civil and Infrastructure Engineering*, 14(5): 357–367.

Prékopa, A. 1971. Logarithmic concave measures with applications to stochastic programming. *Acta Scientiarum Mathematicarum*, 32: 301–316.

Prékopa, A. 1973. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum*, 34: 335–343.

Prékopa, A. 2003. Probabilistic programming. *Handbooks in Operations Research and Management Science*, 10: 267–351.

Prékopa, A. 2013. *Stochastic programming*, volume 324. Springer Science & Business Media.

Santana, P.; Vaquero, T.; Toledo, C.; Wang, A.; Fang, C.; and Williams, B. 2016. Paris: A polynomial-time, risk-sensitive scheduling algorithm for probabilistic simple temporal networks with uncertainty. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 26.

Tsamardinos, I. 2002. A probabilistic approach to robust execution of temporal plans with uncertainty. In *Proceedings of the Hellenic Conference on Artificial Intelligence*, volume 2.

Van Ackooij, W. 2020. A discussion of probability functions and constraints from a variational perspective. *Set-Valued and Variational Analysis*, 28(4): 585–609.

Van Ackooij, W.; Henrion, R.; Möller, A.; and Zorgati, R. 2010. On probabilistic constraints induced by rectangular sets and multivariate normal distributions. *Mathematical Methods of Operations Research*, 71(3): 535–549.

Van Ackooij, W.; Zorgati, R.; Henrion, R.; and Möller, A. 2011. Chance constrained programming and its applications to energy management. *Stochastic Optimization-Seeing the Optimal for the Uncertain*, 291–320.

Vidal, T. 1999. Handling contingency in temporal constraint networks: from consistency to controllabilities. *Journal of Experimental & 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 European Conference on Artificial Intelligence*, volume 12.

Wang, A.; and Williams, B. 2015. Chance-constrained scheduling via conflict-directed risk allocation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Yu, P.; Fang, C.; and Williams, B. 2015. Resolving over-constrained probabilistic temporal problems through chance constraint relaxation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.