Robust Temporal Constraint Network

Hoong Chuin LAU Thomas OU

Melvyn SIM

National University of Singapore, Singapore 119260

lauhc@comp.nus.edu.sg, ouchungk@comp.nus.edu.sg, dscsimm@nus.edu.sg

June 13, 2005

Abstract

In this paper, we propose the Robust Temporal Constraint Network (RTCN) model for simple temporal constraint networks where activity durations are bounded by random variables. The problem is to determine whether such temporal network can be executed with failure probability less than a given $0 \leq \epsilon \leq 1$ for each possible instantiation of the random variables, and if so, how one might find a feasible schedule with each given instantiation. The advantage of our model is that one can vary the value of ϵ to control the level of conservativeness of the solution. From the scheduling perspective, this offers planners the flexibility to generate plans whose minimum makespan vary with ϵ; and where non-renewable resources are available, one can find optimal resources needed to achieve a given makespan within a given confidence level. We present a computationally tractable and efficient approach to solve these RTCN problems. Experimentally, we study the effects the density of temporal constraint networks have on its makespan under different confidence levels. We also apply RTCN to solve the stochastic project crashing problem.

Keywords. Planning and Scheduling, Temporal constraints, Uncertainty.

1 Introduction

In many practical real-world scheduling problems, we are faced with a dynamic environment in which activities often take uncertain time durations. The study of the effects of temporal uncertainties has huge commercial interest, especially in the area of project management where the duration bounds of activities may not be precisely known nor even definable by standard probability distribution functions. Hence, it is in the interest of project managers to determine the precise probability within which a project may be completed by a given deadline.

Similarly, AI planners are interested to check if a plan is temporally consistent. Considering activities with temporal uncertainties, as the number of activities grows, the total number of possible scenarios will grow at an exponential rate. Again, there is a need for an efficient method to verify temporal consistency under uncertain temporal conditions. In most literature, the concern is to determine whether a network is 100% *consistent* (i.e. with *all* temporal constraints satisfied) under uncertainty. This may be too conservative, and often results in wasteful resource allocation to meet the that requirement.

In this paper, we extend the temporal constraint networks literature in two aspects. First, we propose the *robust* temporal constraint network model where each contingent (uncertain) edge is represented by an upper and lower bounds which are themselves random variables. All existing network models can be seen as special cases of this proposed model. Second, we treat the level of consistency as an input parameter. In doing so, we offer the user the flexibility to determine the level of conservativeness in guaranteeing schedule feasibility. We demonstrate experimentally that cost savings can be obtained by adjusting the level of conservativeness.

In the following subsections, we present a literature review on existing temporal constraint network models, followed by a discussion of proactive and reactive scheduling approaches for uncertain temporal activities. We will then survey recent advances on robust optimization. All these considerations lead us to present the main contribution of this paper, which is a unified temporal constraint network model for reasoning and optimization on temporal constraints under uncertainty.

1.1 Literature Review

The idea of a network representation of temporal constraints was first developed by [Dechter et al. 1991]. The widely popular Simple Temporal Constraint Network (STN) may be described as a 4-tuple $\langle V, \mathcal{E}, l, u \rangle$, where V is the set of nodes representing time-points, $\mathcal{E}$ is the set of edge denoting binary constraints on the temporal duration between two time-points, l is the lower bound function $l : \mathcal{E} \rightarrow \Re \cup \{-\infty\}$ and u the upper bound function $u : \mathcal{E} \rightarrow \Re \cup \{+\infty\}$ which maps each edge to a lower and upper bounds of the interval of possible durations. STN can be represented using a directed acyclic distance graph. Each directed edge $(i, j) \in \mathcal{E}$ is labelled by the upper and lower bounded interval $[l_{ij}, u_{ij}]$ which represents the constraint $l_{ij} \leq V_j - V_i \leq u_{ij}$. A tuple $\mathcal{X} = \{x_1, \ldots, x_n\}$ is a *solution* if the assignment $\{V_1 = x_1, \ldots, V_n = x_n\}$ satisfies all the constraints in $\mathcal{E}$. STN is *consistent* if there exists at least one solution.

STN assumes that temporal distance between time points is deterministic. This may not be practical in situations when temporal duration of activities is uncertain and is affected by various external agents. Simple Temporal Network with Uncertainty (STN-u) model [Vidal & Ghallab 1996] attempts to address this issue by allowing uncertain (or uncontrollable) temporal distance between time points within an interval $[l_{ij}, u_{ij}]$. STN-u is a 5-tuple $\langle V, \mathcal{E}, l, u, C \rangle$. $C \subseteq \mathcal{E}$

are set of edges defined as contingent links representing causal processes of uncertain duration. Thus, $\forall c_{ij} \in C$, the temporal distance of c_{ij} may take *any* value inside the interval $[l_{ij}, u_{ij}]$. We define $\mathcal{S}$ as the set of all possible scenarios from the combination of values that $\mathcal{C}$ can take. STN-u is defined to be *strongly controllable* if there exist a solution such that it satisfies all the constraints in $\mathcal{E}$ irrespective of all possible scenarios in $\mathcal{S}$. STN-u is *weakly controllable* if for each possible scenario in $\mathcal{S}$ there exist a solution. STN-u is *dynamically controllable* if there is a solution that may depend on the outcomes of contingent links in the past, but not on those in the future. Consistency algorithms and tractable classes for STN-u were identified in [Vidal & Fragier 1999]. Execution algorithms for STN-u considering the tradeoff between flexibility and efficiency were presented in [Morris & Muscettola 2000]. [Tsamardinos 2002] recently introduced the notion of probabilistic temporal networks where there is no bounding intervals, but instead a set of conditional probability density functions are introduced. They address a scheduling problem that maximizes the probability of correct execution. Their approach does not guarantee that a global optimum will be found. Another AI approach in dealing with uncertain temporal duration is through fuzzy representation. Fuzzy Simple Temporal Constraint Network [Marin et al. 1997] is denoted by $\mathcal{V} = \langle V, C \rangle$ where $\mathcal{C}$ represents the set of fuzzy binary temporal constraints defined between any 2 temporal nodes in $\mathcal{V}$. Every fuzzy binary constraint C_{ij} restricts the possible relative values of V_i and V_j, i.e., $V_j - V_i \leq C_{ij}$. C_{ij} is represented by a possibility distribution Π_{ij} over the continuous time scale $\mathcal{T}$. Thus, there exists a pool of solutions for each degree of possibility. The successful integration of FTCN model into temporal planners [Castillo et al. 2004] to obtain flexible temporal plans shows we can integrate easily network-based model of representing uncertain temporal constraints into AI planners.

1.2 Proactive and Reactive Scheduling

Scheduling of activities under uncertain temporal constraints is an emerging research topic (See, for example, [Policella et al. 2004]). There are two fundamental approaches. A reactive approach locally revises the schedule to account for future events that may violate a temporal constraint in the initial plan. A proactive approach, on the other hand, uses available information about the temporal behavior of the activities and generates off-line a predictive schedule that hedges against temporal constraint violations during the execution of the schedule.

In this work, we propose a proactive strategy with a twist. Instead of generating a fixed schedule that seeks to work well in all scenarios, we are interested, given a probability value, to find a *policy* which is parameterized by the uncertainty variables, such that under any possible projection (i.e. realization of the uncertain variables), the policy will guarantee an executable schedule that is temporally consistent with that probability. Such policy, if it exists, implies that no matter how the uncertain variables are realized, we can always find a feasible schedule where the network will be temporally consistent with the given probability. Hence, by setting the probability value to 1 and fixing the uncertain variables to be constant, we are in effect solving the weak controllability STN-u problem described above.

1.3 Robust Optimization

In the literature, there are several proposals of robust optimization. Some leads to harder problems to solve (see, for example, [Kouvelis & Yu 1999] which shows that their robust version of the shortest path problem is NP-hard). In our opinion, successful applications of robust optimization should have two important characteristics:

(a) Preserving the computational tractability both theoretically and most importantly practically of the nominal problem. From a theoretical perspective it is desirable that if the nominal problem is solvable in polynomial time, then the robust problem is also polynomially solvable.

(b) Being able to find a guarantee on the probability that the robust solution is feasible under some mild assumption of probability distributions.

In view of these criteria, certain robust optimization models have been proposed in immunizing uncertain mathematical optimization against infeasibility while preserving the tractability of the model (See [Ben-Tal & Nemirovsk 2000], [El-Ghaoui et al. 1998] and [Bertsimas & Sim 2003, Bertsimas & Sim 2004]). More recently, [Chen et al. 2004] proposed a robust optimization framework for solving approximate solutions for stochastic optimization models with chance constraints. Their solution approach is based on second-order cone programming. The model permits more sophisticated uncertainty sets that is mapped from random variables.

1.4 Our Contribution

Inspired by the work in [Chen et al. 2004], the motivation of the paper is to develop a unified proactive model that incorporates a broad class of uncertain temporal constraints. More specifically, we propose a model that allows the durations of activities to be bounded on both ends by random variables.

This paper proceeds as follows. We will describe in detail the new Robust Temporal Constraint Network (RTCN) model in the Section 2. In Section 3, we will describe a robust optimization approach in tackling the RTCN problem. Section 4 will describe experimental results on random instances of RTCN, and apply our model to solve the stochastic project crashing problem.

2 Problem Definition

In this section, we formally define the RTCN problem. We denote a random variable by $\tilde{x}$ with the tilde sign. Bold face lower case letters such as $\boldsymbol{x}$ represent vectors and the corresponding upper case letters such as $\boldsymbol{A}$ denote matrices.

Definition 1 *A Robust Temporal Constraint Network is represented by a 5-tuple* $\mathcal{N} = \langle \mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{U}, \epsilon \rangle$

$\mathcal{L}$*: Lower Bound Function that maps each edge* (i, j) *to its lower bound random variable* $\tilde{l_{ij}}$

$\mathcal{U}$*: Upper Bound Function that maps each edge* (i, j) *to its upper bound random variable* $\tilde{u_{ij}}$

$\mathcal{V}$*: Set of nodes representing timepoints*

$\mathcal{E}$*: Set of directed edges where each edge represents the following uncertain temporal constraint:*

$$l_{ij} \leq V_j - V_i \leq u_{ij}, \quad \forall (i,j) \in \mathcal{E}, \tilde{l}_{ij} = l_{ij}, \tilde{u}_{ij} = u_{ij} \tag{1}$$

ϵ*: Threshold probability of temporal constraint violation*

Our proposed model captures uncertainties by labeling each edge with a pair of upper and lower-bound random variables $\left[\tilde{l}_{ij}, \tilde{u}_{ij}\right]$. It is assumed that we know the mean and variance of these random variables, but we do not need to be concerned with the actual probability distribution functions. A *projection* refers to a fixed scenario where all upper and lower bound random variables are instantiated to fixed values. Equation 1 means that under a fixed projection, the duration between V_j and V_i must be within the bound $[l_{ij}, u_{ij}]$.

Definition 2 *A RTCN* $\mathcal{R}$ *is said to be robust controllable (or simply controllable) iff over all possible projections occurring according to their respective*

probability distributions, the probability that there exists an assignment (feasible schedule) $\{V_1 = x_1, \ldots, V_n = x_n\}$ *that satisfies all the constraints in* $\mathcal{E}$ *is at least* $1 - \epsilon$.

We can view ϵ as the upper bound probability that at least one of the temporal constraints specified in (1) is violated. In other words, it defines the desired level of conservativeness we allow for temporal constraint violations. Note that when $\epsilon = 0$, the set of feasible schedules are the ones which satisfy (1) under the worst case scenario. As we increases ϵ, we are in fact increasing the size of the set of feasible robust schedules.

The RTCN Feasibility problem is defined as: given an RTCN instance, determine whether it is robust controllable. This problem is equivalently stated as the problem of finding a *robust policy* such that the probability that a feasible schedule can be derived from the policy with probability $1 - \epsilon$ over the set of all projections. The RTCN Optimization problem is to find a policy that optimizes a given objective function while preserving robust controllability. In this paper, we focus on the optimization of the makespan of the network.

Note that the RTCN problem is a core problem underlying many real-world planning and scheduling problems. For example, in robust scheduling, one is concerned with minimizing the total makespan of the machines where job arrivals or processing times are uncertain. The value of ϵ defines the robustness of the RTCN in the sense that the higher the value of ϵ the more robust the network will be against uncertain temporal constraint violation. However, certain RTCN may not be robust controllable when its constraints are too tight. In such cases, planners may decide to take more risk of temporal constraint violation by increasing ϵ to obtain a robust controllable RTCN.

STN and STN-U are unified under RTCN as shown below:

Observation 1 *Given a binary constraint from STN* $a \leq V_j - V_i \leq b$, *the*

equivalent uncertain bounded temporal constraint is modelled as follows:

$$0 \leq \tilde{l}_{ij} \leq V_j - V_i \leq \tilde{u}_{ij} \leq +\infty$$

$$P\left(\tilde{l}_{ij} = a\right) = \mathrm{P}\left(\tilde{u}_{ij} = b\right) = 1$$

Observation 2 *Given a contingent binary constraint from STN-U, the equivalent uncertain bounded temporal constraint is as follows:*

$$\tilde{l}_{ij} \leq V_j - V_i \leq \tilde{l}_{ij}$$

$$\mathrm{P}(\tilde{l}_{ij}) = \begin{cases} 0 & \textit{for } \tilde{l}_{ij} < a \\ 1 & \textit{for } a \leq \tilde{l}_{ij} \leq b \\ 0 & \textit{for } \tilde{l}_{ij} > b, \end{cases}$$

3 Solution Approach

We will show in this section how a computationally tractable optimization technique due to [Chen et al. 2004] can be applied to solve the RTCN problem.

We first consider an uncertain linear constraint,

$$\tilde{\boldsymbol{a}}'\boldsymbol{x} \geq b \tag{2}$$

in which the vector $\tilde{\boldsymbol{a}}$ is subjected to random perturbation. We are interested in finding feasible solutions such that,

$$\mathrm{P}(\tilde{\boldsymbol{a}}'\boldsymbol{x} \geq b) \geq 1 - \epsilon, \tag{3}$$

which can be computationally challenging. In robust optimization, we solve for solutions under the following "robust counterpart",

$$\boldsymbol{a}'\boldsymbol{x} \geq b \ \forall \boldsymbol{a} \in \mathcal{G}_\Omega, \tag{4}$$

where $\mathcal{G}_\Omega$ is a compact uncertainty set parameterized by Ω, which directly controls the robustness of the solution.

The form of uncertainty set depends on our knowledge of the probability distributions. [Chen et al. 2004] introduced new deviation measures that help us design uncertainty sets from knowledge of probability distributions. These deviation measures will determine the shape and size of the uncertainty set in order to fulfil the chance constraint. We will adopt the same approach in mapping probability distributions of the *primitive* uncertainties to the uncertainty sets used in our robust optimization framework. Hence, in the subsequent paragraphs, we will briefly highlight and explain some of the results of [Chen et al. 2004].

3.1 Perturbation Model for Lower and Upper Bound Random Variables

Instead of arbitrary data dependency for the lower and upper bounds random variables, $(\tilde{l}_{ij}, \tilde{u}_{ij})$, the framework of robust optimization restricts data to be affinely dependent on a collection of independent random variables. In the same spirit, we represent the bound uncertainties as $(\tilde{l}_{ij}, \tilde{u}_{ij})$ affinely dependent on a set of independent random variables, $\{\tilde{z}_k\}_{k=1:N}$ as follows,

$$\tilde{l}_{ij}(\boldsymbol{z}) = l^0_{ij} + \sum_{k=1}^{N} \Delta l^k_{ij} \tilde{z}_k$$
$$\tilde{u}_{ij}(\boldsymbol{z}) = u^0_{ij} + \sum_{k=1}^{N} \Delta u^k_{ij} \tilde{z}_k$$

where l^0_{ij} and u^0_{ij} are the mean values associated respectively with the lower and upper bound random variables and Δl^k_{ij} and Δu^k_{ij}, are the directions of perturbation in respond to the primitive uncertainty $\tilde{z}_k$. Each random variable $\tilde{z}_k$ has mean zero and support in $[-\underline{z}_k, \bar{z}_k]$, $\underline{z}_k, \bar{z}_k > 0$. We call $\{\tilde{z}_k\}_{k=1:N}$, the primitive uncertainties. Hence, under the affine demand uncertainty, we can express the model involving uncertainties solely by using the primitive uncertainties, $\tilde{\boldsymbol{z}}$.

Affine perturbation is certainly a more restricted way of relating multivariate random variables. However, despite its simplicity, it is still a formidable model.

For instance, if the bounds are independent, we will associate every uncertain data with a primitive uncertainty. The bounds may be collectively dependent of certain independent factors such as weather conditions, resource availability, in which case, the parameters will be correlated.

3.2 Deviation Parameter $\tilde{z}$

Operations on random variables with known distributions is generally cumbersome and does not appeal to algorithms design in optimization. For instance, to decide whether the α-quantile of a linear sum of random variables exceeds certain value is usually computationally intractable. The key idea in robust optimization is to approximate probabilistic constraints with tractable formulations by incorporating mild distribution information such as the support and deviation parameters. [Chen et al. 2004] introduce two new parameters: forward and backward deviation parameters, which could be exploit in our proposed approach to obtain nontrivial probability bounds against temporal constraint violation.

Let $\tilde{z}$ be a random variable with zero mean and $M_{\tilde{z}}(s) = \mathrm{E}(\exp(s\tilde{z}))$ be its moment generating function. They define the set of values associated with forward deviations of $\tilde{z}$ as follows,

$$\mathcal{P}(\tilde{z}) = \left\{ \alpha \ : \ \alpha \geq 0, M_{\tilde{z}}\left(\frac{\phi}{\alpha}\right) \leq \exp\left(\frac{\phi^2}{2}\right), \forall \phi \geq 0 \right\}. \tag{5}$$

Likewise, for backward deviations, they define the following set,

$$\mathcal{Q}(\tilde{z}) = \left\{ \alpha \ : \ \alpha \geq 0, M_{\tilde{z}}\left(\frac{-\phi}{\alpha}\right) \leq \exp\left(\frac{\phi^2}{2}\right), \forall \phi \geq 0 \right\}. \tag{6}$$

Furthermore $\mathcal{P}(c) = \mathcal{Q}(c) = \Re_+$ for any constant c.

For known distributions, we define the forward deviation of $\tilde{z}$ as $p^*_{\tilde{z}} = \inf \mathcal{P}(\tilde{z})$ and the backward deviation as $q^*_{\tilde{z}} = \inf \mathcal{Q}(\tilde{z})$. [Chen et al. 2004] show that

these values are no less than the standard deviation. Interesting, under normal distribution, these values coincide with the standard deviation.

Theorem 1 *[Chen et al. 2004] Let $\tilde{z}$ be a random variable with mean zero and standard deviation, σ.*

(a) *$p^*_{\tilde{z}} \geq \sigma$ and $q^*_{\tilde{z}} \geq \sigma$. In addition, if $\tilde{z}$ is normally distributed, then $p^*_{\tilde{z}} = q^*_{\tilde{z}}\sigma$.*

(b) *If $\tilde{z}$ is symmetric in $[-1, 1]$, then $1 \in \mathcal{P}(\tilde{z})$ and $1 \in \mathcal{Q}(\tilde{z})$.*

(c) *If $\tilde{z}$ is bounded in $[-1, 1]$, but not necessarily symmetric, then $1.07 \in \mathcal{P}(\tilde{z})$ and $1.07 \in \mathcal{Q}(\tilde{z})$.*

Although for most distributions we may not be able to obtain close form solutions of p^* and q^*, we can still determine their values numerically. For instance, if $\tilde{z}$ is uniformly distributed over $[-1, 1]$, we can determine numerically that $p^* = q^* = 0.58$. The results of Theorem 1(b,c) suggest that if we do not know the distributions, we can also bound the parameter of deviations from the distribution support.

Model of Uncertainty, U. We assume that the primitive uncertainties of the demands $\{\tilde{z}_k\}_{k=1:N}$, have parameters of deviations, (p_k, q_k) satisfying,

$$p_k \in \mathcal{P}(\tilde{z}_k), q_k \in \mathcal{Q}(\tilde{z}_k)\ \forall k = \{1, \ldots, N\}.$$

The associated uncertainty set is as follows,

$$\begin{aligned} \mathcal{G}_\Omega = \big\{ \boldsymbol{z} \ :\ & \exists \boldsymbol{v}, \boldsymbol{w} \in \Re^N_+, \boldsymbol{z} = \boldsymbol{v} - \boldsymbol{w}, \\ & \|\boldsymbol{P}^{-1}\boldsymbol{v} + \boldsymbol{Q}^{-1}\boldsymbol{w}\|_2 \leq \Omega, -\underline{\boldsymbol{z}} \leq \boldsymbol{v} - \boldsymbol{w} \leq \bar{\boldsymbol{z}} \big\}, \end{aligned} \tag{7}$$

where $\boldsymbol{P} = \text{diag}(p_1, \ldots, p_N)$ and likewise, $\boldsymbol{Q} = \text{diag}(q_1, \ldots, q_N)$.

Theorem 2 *[Chen et al. 2004] Let*

$$y(\boldsymbol{z}) = y_0 + \sum_{k=1}^{N} y_k z_j.$$

The robust counterpart of

$$y(\boldsymbol{z}) \geq 0 \quad \forall \boldsymbol{z} \in \mathcal{G}_\Omega,$$

is equivalent to

$$\left\{ \begin{array}{l} y_0, \ldots, y_N \ : \\ \exists u_0 \in \Re, \boldsymbol{u}, \boldsymbol{t}, \boldsymbol{s} \in \Re^N \\ \quad y_0 \geq \Omega u_0 + \boldsymbol{t}'\bar{\boldsymbol{z}} + \boldsymbol{s}'\underline{\boldsymbol{z}} \\ \quad u_k \geq q_k(y_k - s_k + t_k) \quad \forall k \in \{1, \ldots, N\}, \\ \quad u_k \geq -p_k(y_j - s_k + t_k) \quad \forall k \in \{1, \ldots, N\}, \\ \quad \|\boldsymbol{u}\|_2 \leq u_0 \\ \quad \boldsymbol{t}, \boldsymbol{s} \geq \boldsymbol{0}. \end{array} \right\} \tag{8}$$

In addition, if $y_0, \ldots, y_N$ satisfy the robust counterpart (8), then

$$\mathrm{P}(y(\boldsymbol{z}) \geq 0) \geq 1 - \exp(-\Omega^2/2).$$

3.3 Solving RTCN Problem

In the RTCN Feasibility problem, we are interested to know whether the probability that a temporal constraint network being controllable is at least $1 - \epsilon$. This problem can be cast as a two stage stochastic optimization problem with chance constraint. In the first stage, the primitive uncertainties $\tilde{\boldsymbol{z}} = \boldsymbol{z}$ are realized and hence, the temporal bounds, $\boldsymbol{l}(\boldsymbol{z})$ and $\boldsymbol{u}(\boldsymbol{z})$. In the second stage, we check whether the temporal network is consistent, in other words, whether the set,

$$\chi(\boldsymbol{z}) = \{\boldsymbol{V} : l_{ij}(\boldsymbol{z}) \leq V_j - V_i \leq u_{ij}(\boldsymbol{z}) \quad \forall (i,j) \in \mathcal{E}\},$$

is feasible. The goal is to check whether,

$$\mathrm{P}(\{\chi(\tilde{\boldsymbol{z}}) \neq \emptyset\}) \geq 1 - \epsilon. \tag{9}$$

We can represent a policy as assignment of $V_1(\boldsymbol{z}), \ldots, V_n(\boldsymbol{z})$ for all realization of $\tilde{\boldsymbol{z}} = \boldsymbol{z}$ and the equivalent feasibility problem of (9) is whether there exists such policy $\boldsymbol{V}(\boldsymbol{z})$:

$$\mathrm{P}\left(\left\{\begin{array}{c} V_j(\tilde{\boldsymbol{z}}) - V_i(\tilde{\boldsymbol{z}}) - l_{ij}(\tilde{\boldsymbol{z}}) \geq 0 \\ \forall (i,j) \in \mathcal{E} \\ V_i(\tilde{\boldsymbol{z}}) - V_j(\tilde{\boldsymbol{z}}) + u_{ij}(\tilde{\boldsymbol{z}}) \geq 0 \\ \forall (i,j) \in \mathcal{E} \end{array}\right\}\right) \geq 1 - \epsilon. \tag{10}$$

Unfortunately, finding such policy is generally difficult. The size of the policy is potentially infinite if the underlying distribution is continuous. Even if the distributions are discrete, the size of the policies is generally exponential in N. Hence, to break the bearer of computations, we need to restrict the class of policy to consider. In linear decision rule, we restrict the policy, $\boldsymbol{V}(\tilde{\boldsymbol{z}})$ to be affinely dependent on the primitive uncertainties as follows,

$$\boldsymbol{V}(\tilde{\boldsymbol{z}}) = \boldsymbol{V}^0 + \sum_{k=1}^{N} \boldsymbol{V}^k \tilde{z}_k.$$

Incorporating Theorem 2, and choosing $\Omega = \sqrt{-2\ln(\epsilon/(2|\mathcal{E}|))}$ a sufficient condition for RTCN feasibility is find whether there exists a linear decsion rule, parameterized by

$$\boldsymbol{V}^0, \ldots, \boldsymbol{V}^N,$$

that satisfy the following

$$\left\{\begin{array}{l}
\boldsymbol{V}^0,\ldots,\boldsymbol{V}^N \ : \\
\exists \acute{v}_0^{ij}, \grave{v}_0^{ij} \in \Re, \acute{\boldsymbol{v}}^{ij}, \grave{\boldsymbol{v}}^{ij}, \acute{\boldsymbol{t}}^{ij}, \grave{\boldsymbol{t}}^{ij}, \acute{\boldsymbol{s}}^{ij} \grave{\boldsymbol{s}}^{ij} \in \Re^N, \\
\quad \forall (i,j) \in \mathcal{E} \\
V_j^0 - V_i^0 - l_{ij}^0 \geq \Omega \acute{v}_0^{ij} + \acute{\boldsymbol{t}}^{ij\prime} \bar{\boldsymbol{z}} + \acute{\boldsymbol{s}}^{ij\prime} \underline{\boldsymbol{z}} \\
\quad \forall (i,j) \in \mathcal{E} \\
\acute{v}_k^{ij} \geq q_k(V_j^k - V_i^k - l_{ij}^k - \acute{s}_k^{ij} + \acute{t}_k^{ij}) \\
\quad \forall k \in \{1,\ldots,N\}, (i,j) \in \mathcal{E} \\
\acute{v}_k^{ij} \geq -p_k(V_j^k - V_i^k - l_{ij}^k - \acute{s}_k^{ij} + \acute{t}_k^{ij}) \\
\quad \forall k \in \{1,\ldots,N\}, (i,j) \in \mathcal{E} \\
\|\acute{\boldsymbol{v}}^{ij}\|_2 \leq \acute{v}_0^{ij} \forall (i,j) \in \mathcal{E} \\
V_i^0 - V_j^0 + u_{ij}^0 \geq \Omega \grave{v}_0^{ij} + \grave{\boldsymbol{t}}^{ij\prime} \bar{\boldsymbol{z}} + \grave{\boldsymbol{s}}^{ij\prime} \underline{\boldsymbol{z}} \\
\quad \forall (i,j) \in \mathcal{E} \\
\grave{v}_k^{ij} \geq q_k(V_i^k - V_j^k + u_{ij}^k - \grave{s}_k^{ij} + \grave{t}_k^{ij}) \\
\quad \forall k \in \{1,\ldots,N\}, (i,j) \in \mathcal{E} \\
\grave{v}_k \geq -p_k(V_i^k - V_j^k + u_{ij}^k - \grave{s}_k^{ij} + \grave{t}_k) \\
\quad \forall k \in \{1,\ldots,N\}, \forall (i,j) \in \mathcal{E} \\
\|\grave{\boldsymbol{v}}^{ij}\|_2 \leq \grave{v}_0^{ij} (i,j) \in \mathcal{E} \\
\acute{\boldsymbol{t}}^{ij}, \grave{\boldsymbol{t}}^{ij}, \acute{\boldsymbol{s}}^{ij}, \grave{\boldsymbol{s}}^{ij} \geq \boldsymbol{0} \quad \forall (i,j) \in \mathcal{E}
\end{array}\right\} \tag{11}$$

is feasible.

4 Experimental Results

We conducted 2 experiments on the Robust Temporal Constraint Model. The first is the study of the how the graph density of a temporal constraint network may affect the robust model. In the second experiment, we show an application of RTCN on the problem of project crashing under uncertain project activity duration which is of huge commercial interest. All the algorithms are implemented

in Matlab v6.0 and SDPT3 [Toh et al. 1999] is used as the optimization solver. The platform used is a Pentium 4-2.4GHz processor with 256MB RAM under Windows XP.

4.1 Experimental Study: Impact of Edge Density on Makespan

In this section, we investigate how the edge density (or connection degree) of a temporal network may affect a RTCN model. Particularly, we solve the RTCN Optimization problem of project scheduling with uncertain completion time. The definition of edge density is the ratio of the number of edges in the network over the maximum number of possible edges in a complete graph for a given N number of temporal nodes. Edge density = 1, when the total number of edges in the network is equal to $N(N-1)/2$. Random temporal constraint networks are generated based on the network generation algorithms as described in TimeNetManager [Cesta et al. 1999]. TimeNetManager can generate sets of temporal networks controlled by a set of macro-parameters that characterize the topology and the temporal flexibility of the networks.

For our experiment, we fix the temporal node size = 10, grid ratio = 0.2 and grid density = 1. Each edge is bounded by the interval $[l_{ij}, \infty]$. We introduce uncertainty to the lower-bound variable by replacing l_{ij} with random variable $\tilde{l}_{ij}$ where $\tilde{l}_{ij} = l_{ij}(1 + \tilde{z}_{ij})$. The forward and backward deviation parameters for $\tilde{z}_{ij}$ are p* = q* = 0.1108. The upperbound for $\tilde{z}_{ij}$ is equal to 0.5. We define the robust makespan, Z_ϵ as the time in which the plan can be completed with probability at least $1 - \epsilon$. For each edge density, 10 random temporal networks are generated and we take the mean value of the robust makespan. The robust makespan, $Z_{\epsilon=0.1}, Z_{\epsilon=1}, Z_{\epsilon=5}$ are compared with the makespan for the worst case scenario($\epsilon = 0$). Figure 1 shows the results of our experiment. We can observe that (1) Higher value of ϵ and (2) Lower network edge density offers

more improvements to the make-span.

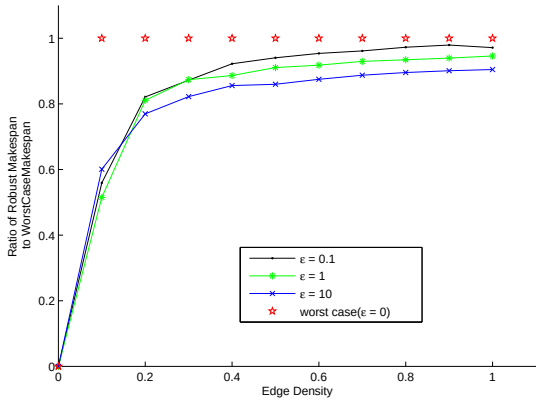

Figure 1: Effect of Density against Robust Makespan

4.2 Application: Robust Project Crashing

Next, we apply the RTCN model to solve the Stochastic Project Crashing problem. Classically, project crashing is done using simulation of a given project activity network. For the project crashing problem, we are given a project activity network with uncertain completion times. For each activity, resources can be added to reduce total completion time of the project. For simplicity, we assume resources are homogeneous. The goal is to find a resource allocation that minimizes the total amount (or cost) of resources added to "crash" the network, i.e. to reduce the makespan to within a given time T. In the stochastic setting, the value ϵ is given and the objective is to minimize the resources needed such that the project can be completed before T with a probability of higher than $1 - \epsilon$. Under RTCN, each activity (i, j) has a lower bound uncertain duration time $\tilde{l}_{ij}$. For each activity, x_{ij} denotes the number of units of resources to reduce a unit duration of the activity. Thus, the completion time for each activity can

be represented by the equation:

$$\tilde{t}_{ij} = (1 + \tilde{z}_{ij})\tilde{l}_{ij} - x_{ij} \tag{12}$$

We run the experiment on a 3-by-12 grid network. Figure 2 shows 4 graphical plots. We compare our results against a simulation of the network with 10,000 instances.

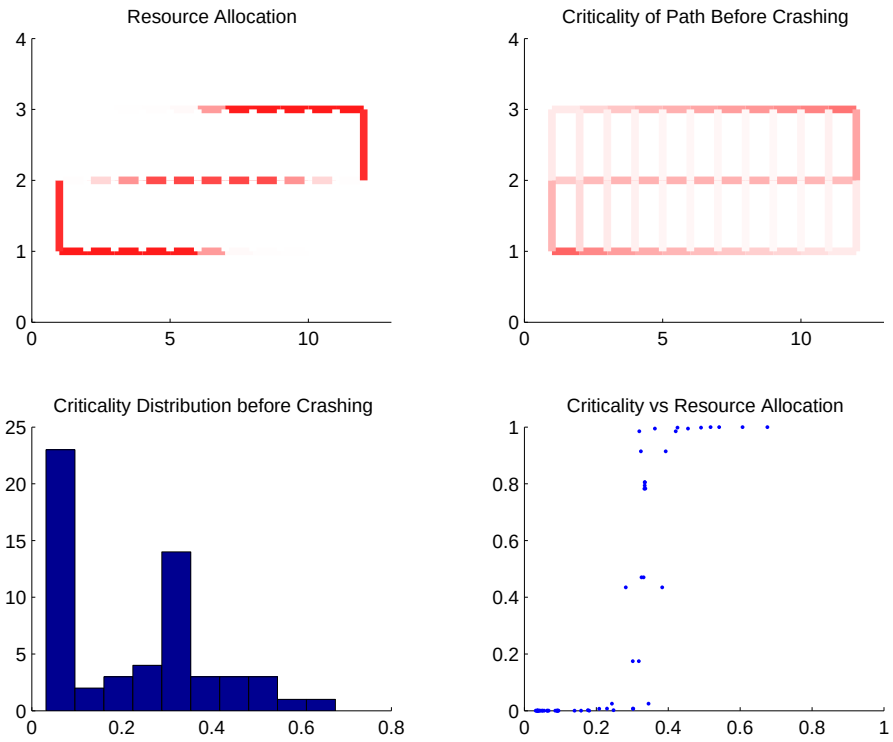

Figure 2: Results of Our Approach against Simulation

The first plot shows the amount of resources added to each activity. Arcs with a darker band denotes that more resources are being added to this activity. The second and the third graphical plots shows the estimated criticality index of an arc and the critical distribution of the network from the simulations. The

criticality index of an arc is calculated by the total number of times the activity lies on a critical path over the 10,000 instances. We show the relationship between the criticality of an arc and the amount of resources added to crash the project using a scatter-plot where we plot each activities' resource allocation against its criticality index. From the figures, we can see a remarkable match between the activities with more crashing as proposed by our robust model and the criticality of the activities derived from simulation.

References

[Ben-Tal & Nemirovsk 2000] Ben-Tal, A. and Nemirovski, A. (2000): Robust solutions of Linear Programming problems contaminated with uncertain data, *Math. Progr.*, 88, 411-424.

[Bertsimas & Sim 2003] Bertsimas, D. and Sim, M. (2003): Robust Discrete Optimization and Network Flows, *Math. Progr.*, 98, 49-71.

[Bertsimas & Sim 2004] Bertsimas, D. and Sim, M. (2004): Price of Robustness, *Oper. Res.*, 52(1), 35-53.

[Castillo et al. 2004] Castillo L., Fdez.-Olivares J., Gonzalez A. (2004): Integration of fuzzy scheduling into a planning framework, *ECAI 04 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems*.

[Cesta et al. 1999] Cesta, A., Oddi A., and Sussi A. (1999): TimeNetManager - A Software Tool for Generating Random Temporal Networks, *Proc. AI*IA*, 143-154.

[Chen et al. 2004] Chen X., Sim M., and Sun P. (2004): A Robust Optimization Perspective to Stochastic Programming. Working Paper. Available via https://netfiles.uiuc.edu/xinchen/www/papers/

[Dechter et al. 1991] R. Dechter, I. Meiri, and J. Pearl (1991): Temporal Constraint Networks, *Artificial Intelligence*, 49, 91-95.

[El-Ghaoui et al. 1998] El-Ghaoui, L., Oustry, F., and Lebret, H. (1998): Robust solutions to uncertain semidefinite programs, *SIAM J. Optim.*, 9, 33-52.

[Hermann 1999] Hermann J. W. (1999): A Genetic Algorithm for Minimax Optimization Problems, *Proc. Congress on Evolutionary Computation 1999*, 2, 1099-1103.

[Jensen & Hansen 1999] Jensen M. T. and Hansen T. K. (1999): Finding worst-case flexible schedules using coevolution, In *Proc. GECCO 2001*, 1144-1151.

[Kouvelis & Yu 1999] Kouvelis P. and Yu G. (1999): Robust discrete optimization and its application, *Kluwer Academic Publishers.*

[Marin et al. 1997] Marin R. , Cardenas M. A., Balsa M., and Sanchez J.L. (1997): Obtaining solutions in fuzzy constraint networks, *International J. Approximate Reasoning*, 16, 261-288.

[Morris & Muscettola 2000] Morris, P., and Muscettola, N. (2000): Execution of temporal plans with uncertainty, *AAAI-2000*, 491-496.

[Policella et al. 2004] Policella N., Smith S., Cesta A., and Oddi A. (2004): Generating Robust Schedules through Temporal Flexibility, *Proc. ICAPS 2004.*

[Vidal & Ghallab 1996] Vidal, T. and Ghallab, M (1996): Dealing with uncertain durations in temporal constraint networks dedicated to planning, *Proc. ECAI 1996*, 48-52.

[Vidal & Fragier 1999] Vidal, T. and H. Fragier (1999): Handling consistency in temporal constraint networks: from consistency to controllabilities, *J. Experimental & Theoretical AI*, 11, 23-45.

[Toh et al. 1999] Toh, K.C. , Todd, M.J. , and Tütüncü, R.H. (1999): SDPT3- A Matlab Software package Programming, *Optimization Methods and Software*, 11, 545-581.

[Tsamardinos 2002] Tsamardinos I. 2002: A Probabilistic Approach to Robust Execution of Temporal Plans with Uncertainty. In Vlahavas and Spyrpoulos (Eds): *SETN 2002, Lecture Notes in AI (2308)*, 97-108.