

Measuring and Optimizing Durability against Scheduling Disturbances

Joon Young Lee,* Vivaswat Ojha,* James C. Boerkoel Jr.

Human Experience and Agent Teamwork Lab (heatlab.org)

Harvey Mudd College

Claremont, California 91711

{joolee, vmojha, boerkoel}@hmc.edu

Abstract

Flexibility is a useful and common metric for measuring the amount of slack in a Simple Temporal Network (STN) solution space. We extend this concept to specific schedules within an STN's solution space, developing a related notion of durability that captures an individual schedule's ability to withstand disturbances and still remain valid. We identify practical sources of scheduling disturbances that motivate the need for durable schedules, and create a geometrically-inspired empirical model that enables testing a given schedule's ability to withstand these disturbances. We develop a number of durability metrics and use these to characterize and compute specific schedules that we expect to have high durability. Using our model of disturbances, we show that our durability metrics strongly predict a schedule's resilience to practical scheduling disturbances. We also demonstrate that the schedules we identify as having high durability are up to three times more resilient to disturbances than an arbitrarily chosen schedule is.

Introduction

When carrying out a set of tasks, an autonomous system is likely to face unforeseen disturbances. For instance, in a factory setting, there “are many disturbances that can upset a plan, including machine failures, processing time delays, rush orders, quality problems and unavailable material” (Vieira, Herrmann, and Lin 2003). With many machines involved, creating a new schedule from scratch every time a disruption occurs is expensive, making it preferable to operate with schedules that reduce the need for active rescheduling. Thus, the ability to prepare schedules that are *durable*—schedules that do not easily break even when faced with unexpected disturbances—is desirable.

Previous work has examined the notion of flexibility with respect to scheduling *problems* defined as networks of temporal constraints over events. (Hunsberger 2002; Policella et al. 2009; Wilson et al. 2014; Huang et al. 2018). While this work is useful in classifying the flexibility of *solution spaces*, it does not provide information about *particular schedules* within a given problem. In this sense, *durability* can be viewed as flexibility defined on individual schedules.

It is desirable in applications that require schedules that are robust despite a dynamic environment and imperfect agents.

In this paper, we motivate the practical need for a measure of durability against scheduling disturbances and propose several durability metrics for measuring an individual schedule's resistance to disturbances. We contribute two new methods for finding schedules within an STN's solution space that are highly durable. We also develop a novel empirical framework for modeling and testing a schedule's ability to withstand practical scheduling disturbances. Finally, we empirically validate that our new durability metrics are highly predictive of how resilient a schedule is to practical, real-world disturbances and also demonstrate that our methods for finding maximally durable schedules lead to schedules that are up to three times more resilient to disturbances than the average schedule.

Background: Simple Temporal Networks

A simple temporal network (STN) is a set of events T , with events labeled $t_0, t_1, t_2, \dots, t_n$, coupled with a set of m constraints C where each constraint is of the form $t_j - t_i \leq c_{ij}$ (Dechter, Meiri, and Pearl 1991). We consider the event t_0 to be the zero timepoint, fixed to occur at time zero, with all other events occurring relative to it.

A *schedule* is an assignment of times to each of the events in the STN, and is a *solution* of the STN if all constraints are satisfied. A common technique to compute solutions involves representing an STN as a *distance graph*. This is a weighted, directed graph, where each event t_i is a node in the graph while each constraint is represented by a directed edge from t_i to t_j with weight c_{ij} . An STN can be made *minimal* by applying a shortest-path algorithm to its distance graph. In the resulting STN, the edge between two nodes is the shortest path between them in the distance graph. The resulting edge weights represent the exact range of times that is allowed to elapse between each pair of events in the constraint graph and represents the space of possible solutions.

An example STN is shown in Figure 1. Here, each edge/interval pair represents a pair of constraints between two events. This STN describes a problem where two events t_1 and t_2 must happen at least zero but no more than 10 minutes after t_0 , and t_2 must happen no earlier than t_1 .

The solution space of an STN can also be viewed as a convex polytope (Huang et al. 2018), where the number of

*Primary authors listed alphabetically but contributed equally.
Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

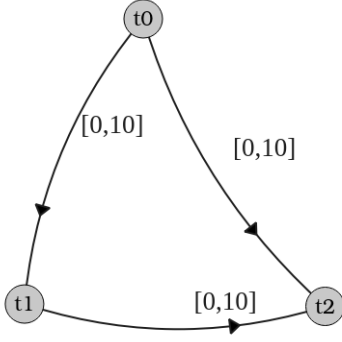


Figure 1: Distance graph representation of an example STN

events in an STN other than the zero timepoint specifies the dimension of the polytope. We later use this geometric interpretation of STNs to find points that we expect to have high resistance to disturbance. The STN described previously is shown geometrically in Figure 2.

In this view of an STN, the square bounded by the dashed lines represents the solution space in $\mathbb{R}^2$ if there were no constraint between the two events, whereas the shaded triangle represents the true solution space after accounting for that constraint. The constraints bounding t_1 are the lines $t_1 = 0$ and $t_1 = 10$, the constraints bounding t_2 are the lines $t_2 = 0$ and $t_2 = 10$, and the inter-event constraint is represented by the diagonal line $t_2 = t_1$. Note that every point inside the shaded triangle is a valid schedule for completing the tasks t_1 and t_2 , e.g., both tasks being executed as soon as possible is represented as the point $(0, 0)$ in the plot above.

Previous work has identified different notions of flexibility for STNs, all of which loosely attempt to capture the flexibility to maneuver a schedule within its constraints. These often attempt to aggregate the available slack moving events around in an STN’s solution space. A survey and analyses of available flexibility metrics are presented by Wilson et al. (2014) and Huang et al. (2018).

Schedule Durability

If all uncertainty can be characterized prior to execution (i.e., *known unknowns*), then an agent can simply compute a strategy that represents its best response to control for this uncertainty (e.g., Morris 2014, Lund et al. 2017). This, in turn, obviates the need for additional flexibility in the network. In practice, however, even within the STN there are often additional sources of uncertainty that are not captured by a formal model of temporal uncertainty. This motivates the second type of uncertainty that acts as the motivating focus for this paper, unanticipated scheduling disturbances (i.e., *unknown unknowns*)—scheduling errors or disruptions that arise or are realized during execution but may not be accurately modeled or known prior to execution. The fact that flexible STNs are generally preferred to inflexible ones is an implicit recognition that such uncertainty exists and points to the possibility of more precisely characterizing how durable individual schedules are to such disturbances.

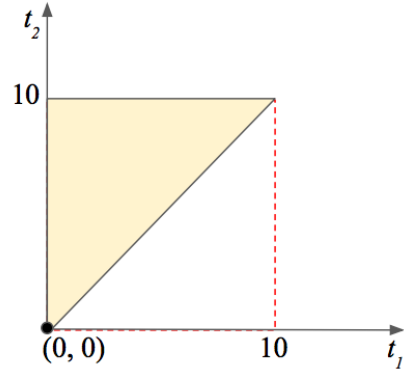


Figure 2: Geometric view of the STN from Figure 1

Such unforeseen scheduling disturbances may arise due to scheduling imprecision based on agents’ limitations or based on other inaccuracies in how faithfully an STN models the real world.

We define the *durability* of a schedule as its ability to withstand disruptions within the constraints of the original STN. A durability metric should vary depending on the characteristics of, and allow comparisons between, individual schedules within an STN solution space. The geometric perspective of an STN solution space provides an intuitive way to think about durability. Given a schedule, which is a single point inside the STN solution space, its distance from the boundaries provides a measure of the point’s ability to handle disturbances while remaining within the STN constraints, and thus remaining a valid solution.

Durability Metrics

There are a number of different ways to measure the distance between a schedule and the boundary of the solution space, which we consider a heuristic for durability. We propose two: minimum distance to a boundary and expected distance to any boundary.

minDist The *minDist metric* computes the perpendicular distance of a point to its nearest boundary as its estimate of its durability. Note that the largest N -dimensional hypersphere centered on this point will be tangent to the closest boundary. Thus, this metric is equivalent to the radius of that hypersphere, and can be considered a measure of the worst case scenario for the schedule in terms of how much disturbance the schedule can take. The metric ignores the structure of the STN outside of the boundary closest to the schedule and thus would underestimate the potential durability of points which are close to one boundary but far away from all others.

expDist The *expDist metric* attempts to estimate the expected distance of a schedule to all boundaries by computing the geometric mean over the perpendicular distances to each of the boundaries. This metric gives higher values for points further away from the boundaries and approaches zero for points close to any of the boundaries, aligning with the idea

of durability as a point’s ability to remain within the boundaries despite disturbances. It serves as an approximation of the expected case, given that it accounts for disturbances towards every boundary equally.

Note that we can compute the perpendicular distance from a schedule to a given constraint boundary by first calculating, given a constraint of the form $t_j - t_i \leq c_{ij}$, the amount of leeway in that constraint with the equation $dist \leftarrow c_{ij} - t_j + t_i$. We then check if the constraint is in between two events that are both not the initial event, t_0 . If so, we divide the calculated leeway value by $\sqrt{2}$, since the calculated value for two nonzero events will give the distance to a boundary, but not the perpendicular distance to that boundary. To find the min/exp distance, then requires iterating over the $O(n^2)$ possible boundaries for a total computational complexity of $O(n^2)$, where n is the number of events.

Maximally Durable Schedules

Next, we describe two candidate approaches for finding maximally durable schedules by approximating the “center” of the solution space.

Chebyshev Center The largest inscribed sphere that fits within an N -dimensional polytope can be found in pseudo-polynomial time by formulating the problem as a linear program (Murty 2009). The center of this sphere, known as the Chebyshev center, maximizes the minimum distance to any boundary (i.e., the point with the highest minDist value). Thus, the Chebyshev center optimizes for the minimum distance, allowing for robustness under worst case conditions.

Centroid In a polytope, in this case the STN solution space, the centroid is defined as center of mass, which is essentially the mean position of all the points in the polytope. When disturbances of all kinds are equally likely, the centroid is expected to be the most durable schedule. Computing the centroid of an N -dimensional polytope is $\#P$ hard (Rademacher 2007). However, we can approximate the centroid by averaging randomly sampled schedules drawn from a uniform distribution over the the STN solution space by using a Hit-and-Run sampling method as described by Bélisle, Boneh, and Caron (1998). In practice, we found that 500 samples provided reasonable convergence.

Empirical Evaluation

In this section, we first present two new Monte-Carlo-style methods for simulating how far a schedule can be perturbed without being invalidated and then discuss how we use these to empirically evaluate our durability metrics and candidates for maximally durable schedules.¹

Empirically Modeling Unknown Disturbances

We empirically model the fact that the STN models themselves may contain inaccuracies that are only discovered at execution time by randomly selecting one of the edges of a

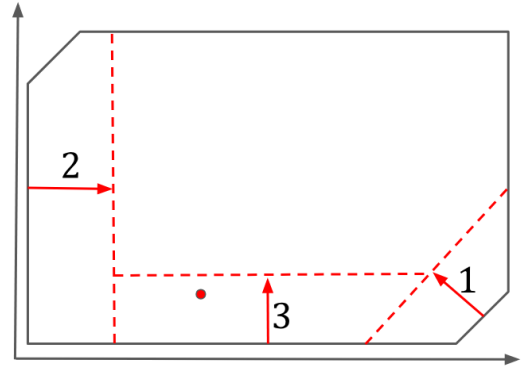


Figure 3: Illustration of our **Random Shave** model *unknown disturbances*.

minimized STN and tightening its bound by one unit. We coin this as a **random shave** because we can view this process geometrically as selecting one of the surfaces of the STN solution space and shaving it to be one unit tighter. We repeat this process until the schedule violates any of the (tightened) constraints (i.e., until the point representing the schedule has fallen out of the polyhedron). We record the number of tightened constraints it took to reach this point, as depicted in Figure 3.

Each tightening of a constraint represents a possible modeling inaccuracy (e.g., the arrival of a resource is delayed or a deadline comes unexpectedly early). Thus, a schedule that, on average, withstands a higher number of such changes while remaining valid, is more durable to such errors in the model. Although it is possible for model inaccuracies to occur in the opposite direction and loosen the constraints, as explored in previous work (Casanova, Pralet, and Lesire 2015; Planken, de Weerd, and Yorke-Smith 2010), we focus solely on tightening. In practice, we found that relaxing constraints, which increases the size of the solution space of the STN, led to similar high-level trends, but served to obfuscate trends by adding an additional source of noise.

Our next empirical model captures the fact that it can be hard to predict exactly when and to what degree a scheduling disturbance will occur. We model this using a **random walk** that randomly assigns the proportion of the displacement that takes place in a given dimension. Geometrically, if each of the events happens at a different time, the point representing the schedule is displaced by a certain magnitude, which we normalize to one standard unit so that we can measure the total displacement. We accomplish this by choosing a vector from a uniform distribution around an N -dimensional unit sphere according to the method detailed by Muller (1959), then using the coordinates of that N -dimensional unit vector as the displacements for each of our events. We also considered versions that, e.g., perturbed individual events or walked non-uniformly, but in practice found these did not lead to qualitative differences compared to the random walk described in this paper.

To evaluate a point’s ability to withstand execution imprecision, we repeat these displacements, checking each time

¹All code and problem instances are available for download from <https://github.com/HEATlab/durability>.

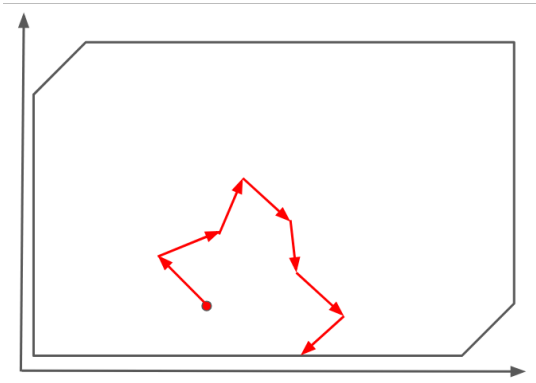


Figure 4: Illustration of our **Random Walk** model *unknown disturbances*.

whether the new point remains a valid schedule. This can be visualized as a "random walk", which traces out a path from the original starting point to a boundary of the STN's solution space as shown in Figure 4. As soon as it violates a constraint, we record the number of displacements it took for the point to do so. Next, we turn to measuring how susceptible schedules are to unknown disturbances and selecting schedules that are maximally durable.

Empirical Setup

Our empirical testing relies on generating STNs based on a variation of Hunsberger's generation method (Hunsberger 2002). Our generator takes as input a number of events, n , and an upper limit on constraint bounds, B . For each of the n^2 pairs of events, i and j , we set c_{ij} to 0 when $i = j$ and to a uniformly-chosen value between 0 and B otherwise. The resulting network is checked for feasibility and is discarded and regenerated from scratch if infeasible. We generated 30,000 randomly structured STNs in this way, for $n \in \{2, 3, \dots, 10\}^2$ and $B = 50$.

Unless otherwise noted, all values that we report represent the average across 100 randomly chosen schedules in each of our 30,000 randomly-generated STNs. To validate our flexibility metrics, we use our empirical *random shave* and *random walk* models, as discussed previously. Both of these models are Monte Carlo simulations that report the total number of perturbations that a schedule survives. For both of these measures, we report the average across 100 simulations.

Empirical Evaluation of Durability Metrics First, we explore correlation between our metrics and resistance to disturbances for schedules within a given STN. For each of the 30,000 random STN instances that we generate, we uniformly sample the solution space (using the same Hit-and-Run algorithm we used for approximating the centroid) to generate 100 feasible schedules within that STN and measure how durable each schedule is using our two empirical

²We considered scaling to larger n , but found that for $n > 20$ we encountered an inverse curse-of-dimensionality, where the solution spaces degenerated as an artifact of the random STN generator

	minDist	expDist	Any STN Flex Metric
Random Shave	0.790	0.762	—
Random Walk	0.873	0.795	—

Table 1: Average Pearson Correlation Coefficients for schedules within STN instances

models of scheduling disturbances. For each STN instance, we compute how predictive (using a Pearson correlation coefficient) each of our durability metrics was with how long the schedule resisted our empirical model of scheduling disturbances.

Both the minDist and expDist durability metrics have strong positive correlations with resistance to both types of disturbances. This shows that they are good measures of how much execution imprecision and how many model inaccuracies a schedule can withstand before being rendered invalid. It is not surprising that these metrics, which are both measures of the distance to boundaries, would be strongly correlated with the measures of durability from our empirical models of disturbances. Perhaps more surprisingly, we also tested a third metric, maxDist, which computed the maximum distance to a boundary, but this proved uselessly optimistic, yielding *negative* correlations. Note that because flexibility depends only on the constraints of the original problem as a whole, and not on characteristics of individual solutions to the problem, a flexibility's metric does not change within a given STN, and thus yields no correlation to a schedule's resilience to disturbances. However, we suspect flexibility metrics would be superior at evaluating the flexibility to actively dispatch schedules by acting as a guide for agents making online scheduling decisions. In summary, both the minDist and expDist proved highly durable to simulated disturbances, and overall, minDist had slightly better performance.

Empirical Performance of STN Centers

In order to test whether our candidates for durable schedules indeed have high resistance to disturbances, we calculated the durability metric values for each of these centers in every STN and compared it against the average of the values for 100 random points in the STN. We then repeated the same process for both the RS and RW measures of resistance to disturbances. The two candidates for maximally durable schedules that we identified, the Chebyshev center and the centroid, both yielded higher durability metric values and resistance to RS and RW models of disturbances than randomly selected schedules in expectation. Both candidates showed consistent, robust advantages over randomly selected schedules. These advantages persisted across every STN instance that we tested.

Table 2 shows the ratio of each of our candidate centers compared to the expected performance of a randomly selected schedule across the minDist durability metric (our best performing durability metric) and also resistance to RS and RW disturbances. In expectation, our centers were more robust than an arbitrarily chosen schedule by an average of

	minDist	Random Shave	Random Walk
Chebyshev	4.69	2.70	3.37
Centroid	3.42	3.87	3.45

Table 2: Center Performance Relative to Random Schedules

306% according to minDist, 229% according to RS, and 241% according to RW. These results strongly point towards the usefulness of using durability metrics in choosing target schedules that are resistant to unknown disturbances.

We were surprised at how remarkably consistent both center candidates performed across our various measures of durability—Chebyshev outpaced arbitrary schedules by an average of 259% and Centroid by 258%. Further, no center consistently dominated the other, in fact, each had at least one metric by which they would be deemed the best.

This points to a larger phenomenon—any schedule that keeps a safe distance from boundaries seems to perform reasonably well. In fact, we explored a variety of different STNs properties (e.g., dimensional, structure, etc.) looking for types of scenarios where one center would outperform others, and found no clear winners. This affords agents with the opportunity to optimize for other objectives when selecting for schedules as long as they are playing a safe distance from the edge.

Discussion

In this paper, we extend the notion of flexibility for Simple Temporal Networks to schedules within these networks by introducing a new metric called schedule durability. We define the notion of schedule durability using a geometric interpretation of STNs, whereby a schedule’s distance to the network’s boundaries approximates its resilience towards unforeseen disturbances. This motivates a number of durability metrics relying on a schedule’s distance to boundaries. We also identify two geometric centers within a STN’s solution space that look to provide optimal resilience against unknown disturbances. We practically motivate the presence of real-world scheduling disturbances and translate these to Monte Carlo style tools for empirically simulating and evaluating a schedule’s resistance to such disturbances. Our empirical analysis show that durability metrics provides useful, new information when assessing which schedule is most likely to resist unknown disturbances by demonstrating they are strong predictors of the performance of our Monte Carlo models of resistance to disturbances. We also demonstrated that geometric centers provide three times more resistance to unforeseen disturbances than arbitrarily chosen schedules.

Acknowledgments

Funding for this work was graciously provided by the National Science Foundation under grant IIS-1651822. Thanks to the anonymous reviewers, HMC faculty, staff and fellow HEATlab members for their support and constructive feedback.

References

- Bélisle, C.; Boneh, A.; and Caron, R. J. 1998. Convergence properties of hit-and-run samplers. *Stochastic Models* 14(4):767–800.
- Casanova, G.; Pralet, C.; and Lesire, C. 2015. Managing dynamic multi-agent simple temporal network. In *Proc of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS-15)*, 1171–1179.
- Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. In *Knowledge Representation*, volume 49, 61–95.
- Huang, A.; Lloyd, L.; Omar, M.; and Boerkoel, J. C. 2018. New perspectives on flexibility in simple temporal planning. In *Proc. 28th Int. Conf. on Automated Planning and Scheduling (ICAPS-2018)*, 123–131.
- Hunsberger, L. 2002. Algorithms for a temporal decoupling problem in multi-agent planning. In *Proc. 18th National Conference on AI (AAAI-02)*, 468–475.
- Lund, K.; Dietrich, S.; Chow, S.; and Boerkoel, J. 2017. Robust execution of probabilistic temporal plans. In *Proc. of the 31st National Conference on Artificial Intelligence (AAAI-17)*, 3597–3604.
- Morris, P. 2014. Dynamic controllability and dispatchability relationships. In *Proc. of the International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR)*, 464–479. Springer.
- Muller, M. E. 1959. A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM* 2(4):19–20.
- Murty, K. G. 2009. Ball centers of special polytopes. Technical report, Dept. of IOE, University of Michigan, Ann Arbor, MI-48109, USA.
- Planken, L.; de Weerd, M.; and Yorke-Smith, N. 2010. Incrementally solving stns by enforcing partial path consistency. In *Proc. 20th Int. Conf. on Automated Planning and Scheduling (ICAPS-2010)*, 129–136.
- Policella, N.; Cesta, A.; Oddi, A.; and Smith, S. F. 2009. Solve-and-robustify. *Journal of Scheduling* 12(3):299.
- Rademacher, L. A. 2007. Approximating the centroid is hard. In *Proc. of the Twenty-third Annual Symposium on Computational geometry - SCG 07*.
- Vieira, G. E.; Herrmann, J. W.; and Lin, E. 2003. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling* 6(1):39–62.
- Wilson, M.; Klos, T.; Witteveen, C.; and Huisman, B. 2014. Flexibility and decoupling in simple temporal networks. *Artificial Intelligence* 214:26–44.