Scheduling Live Interactive Narratives with Mixed-Integer Linear Programming

Sasha Azad
North Carolina State University
8 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Boyang Li
Baidu Research
48 PUBLICATIONS **395** CITATIONS
SEE PROFILE

Proceedings, The Thirteenth AAAI Conference on
Artificial Intelligence and Interactive Digital Entertainment (AIIDE-17)

Scheduling Live Interactive Narratives with Mixed-Integer Linear Programming

Sasha Azad,[1] Jingyang Xu,[2] Haining Yu,[2] Boyang Li[1]
[1]Disney Research
[2]Decision Science, Walt Disney Parks and Resorts
sasha.azad@disneyresearch.com, {jingyang.xu, haining.yu, boyang.li}@disney.com

Abstract

A live interactive narrative (LIN) is an experience where multiple players take on fictional roles and interact with real-world objects and actors to participate in a pre-authored narrative. Temporal properties of LINs are important to its viability and aesthetic quality and hence deserve special design consideration. In this paper, we tackle the largely overlooked problem of scheduling a multiplayer interactive narrative and propose the Live Interactive Narrative Scheduling Problem (LINSP), which handles reasoning under temporal uncertainty, resource scheduling, and non-linear plot choices. We present a mixed-integer linear programming formulation of the problem and empirically evaluates its scalability over large narrative instances.

Introduction

The holy grail of AI research in interactive storytelling (Louchart and Aylett 2004; Roberts and Isbell 2008; Arinbjarnar, Barber, and Kudenko 2009; Riedl and Bulitko 2013) is a narrative world that cannot be distinguished from the real world. Arguably, the closest form that exists today is a live interactive narrative (LIN), a role-playing game where players can interact with human actors at real-world locations, within an overarching narrative that blends both real and virtual elements (Shilkrot, Montfort, and Maes 2014). Compared to traditional live action role-playing, where plots can be created by game masters dynamically, LINs usually feature a predefined story with a small number of plot choices. The rigid structure simplifies the design of the narrative, allowing for the preparation of high-quality materials.

A prominent example of a LIN is the second phase of the Alternate Reality Game (ARG) *Conspiracy For Good* (Stenros et al. 2011). After spending a few months collecting information online, players gathered on London's streets for four action scenes over four days, involving a large number of human actors. Interactive theater productions, such as *Sleep No More* (Brantley 2011), offer another variant where players can explore in an indoor environment with actors but cannot influence plot decisions. The game *Bad News* (Samuel et al. 2016) situates improvisational acting in a computationally simulated town in an one-on-one interactive setting.

Despite artistic and commercial successes, LINs remain a niche form of entertainment, partly due to the cost of designing and operating them. Unlike virtual experiences, real-world resources such as actors, rooms and props incur significant operating cost. Interactions between actors and players may be of uncertain lengths, which is a challenge to a LINs' operation. The initial authoring cost of LINs is high; adapting them to a different environment is not straightforward.

We propose that an AI scheduling algorithm can mitigate these challenges and encourage wide adoption of LINS. The algorithm can optimize the utilization of resources in order to reduce idle time and improve efficiency. It simplifies the adaptation of LINs to different time and resource constraints, which improves reusability and amortizes the initial development cost (Hansen et al. 2013).

The contributions of this paper are as follows.

- We propose the Live Interactive Narrative Scheduling Problem (LINSP), which consolidates the Simple Temporal Problem with Uncertainty (STPU) (Vidal and Fargier 1999; Cui et al. 2015), the Resource-Constrained Project Scheduling Problem (RCPSP) (Artigues, Demassey, and Nron 2008), as well as plot choices, which are specific to interactive narratives.
- We formulate the problem as mixed-integer linear programming (MiLP), which allows the use of plot choices to accommodate temporal uncertainty in scheduling.
- Our empirical evaluation suggests the MiLP problems for typical-shaped LINs can be solved relatively quickly with modern solvers, despite the fact that LINSP is NP-hard.

To our knowledge, this is the first attempt to automatically schedule a live interactive narrative.

Related Work

Temporal constraints and properties have been studied in interactive storytelling and story generation. Porteous et al. (2011) employed temporal planning techniques in story generation with well-defined event lengths. Winer et al. (2016) proposed temporal properties of story discourse that affect comprehension. Story generation with multiple human players in virtual worlds are also investigated (Fairclough and Cunningham 2003; Riedl et al. 2011; Tomai 2012). Computational improvisational theater (Hodhod and Magerko

2014; Martin, Harrison, and Riedl 2016) is another line of work related to a real-world stage, but existing works focus on virtual characters, for which temporal and resource constraints are not crucial.

The role of time in the design of ARGs and LINs have also beend discussed (Benford and Giannachi 2008). Tychsen and Hitchens (2008) proposed a theoretical model for the role of time in multiplayer role playing games. Hansen et al. (2013) discussed reusability and adaptability for ARGs and suggested that reliance on live events and human actors can reduce reusability. In this paper, we do not directly address the design and content of LINs, but our work can be used to schedule a well designed LIN to a different set of temporal constraints, thereby improve its reusability. Similar to our work, MacVean et al. (2011) proposed an algorithm for adapting ARGs to different geographical regions.

To the best of our knowledge, in this paper we present the first scheduling algorithm that supports LINs. Our technique differs from existing scheduling approaches by using plot choices in LINs to accommodate temporal uncertainty and resource constraints.

Background

We review two problems that respectively handle temporal uncertainty and resource usage, upon which our problem is built.

Simple Temporal Problem with Uncertainty

The Simple Temporal Problem with Uncertainty (STPU) (Vidal and Fargier 1999) is a constraint satisfaction problem for events with temporal uncertainty. An STPU is defined on a directed graph $G = \{V, E\}$, where V is a set of vertices representing time points and E is a set of directed edges that represent bounded intervals between time points. Every edge is associated with a lower and an upper limit on its duration. STPU further differentiates between two types of intervals:

- A *requirement edge* (Req) is an interval whose length is controlled by the system.
- A *contingent edge* (Ctg) is an interval whose length is decided by uncontrollable external factors but always within pre-specified upper and lower limits.

We use the two types of intervals to model player interactions, which introduce temporal uncertainty into the narrative. For example, the time it takes for a player to find a location or an object is usually uncontrollable. STPU utilizes controllable events to compensate for this uncertainty and quantify the amount of temporal control that a drama manager has over the narrative.

We refer to an assignment to the lengths of all controllable intervals as a *solution* and the lengths of all uncontrollable intervals as a *situation*. The feasibility problem asks if we can find a solution such that, no matter what the situation turns out to be, all temporal constraints are satisfied. This notion can be further refined into three types of controllability criteria, which we briefly describe below.

- *Strong controllability* implies that we can find an optimal solution that is valid for all possible situations.

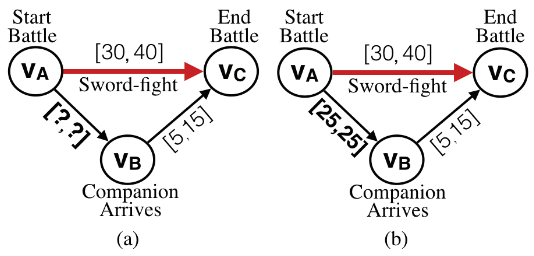


Figure 1: An example STPU with three time points, v_A, v_B, and v_C. The contingent constraint c_{AB} is shown with a red arrow.

- *Weak controllability* implies that for any completely observed situation, we can find a corresponding solution that satisfy all constraints. However, this is unrealistic because the solution depends on the knowledge of all contingent intervals, including future events.
- *Dynamic controllability* implies that we can make decisions based only on past observations of the situation to find valid solutions. This is the most useful definition for scheduling and the problem can be solved in polynomial time (Morris, Muscettola, and Vidal 2001).

As an illustration, Figure 1(a) shows the relationships between three time points in an interactive narrative. The protagonist is fighting a dragon while a companion of the protagonist arrives to help. The fight starts at point v_A, ends at v_C, and has an uncertain duration of 30 to 40 minutes depicted by the contingent edge e_{AC}. We want the protagonist to experience both individual and team fighting, so we want the companion's arrival (v_B) to be 5 to 15 minutes before the battle ends.

To achieve dynamic controllability, the decision on e_{AB} cannot depend on the observation of e_{AC} because C happens after B. Figure 1(b) shows a new set of upper and lower bounds on e_{AB}, $[25, 25]$, which satisfies all the temporal constraints. The solution is valid regardless of the exact length of e_{AC}. For instance, if the length of e_{AC} is 30, then e_{BC} is 5. If e_{AC} is 40, e_{BC} is 15. Intuitively, solving an STPU means adjusting the controllable intervals to cope with the uncertainty posed by contingent edges.

Resource Constrained Scheduling

A live interactive narrative may utilize real-world resources such as props, actors, and rooms. The resource-constrained project scheduling problem (RCPSP) (Artigues, Demassey, and Nron 2008; Artigues, Michelon, and Reusser 2003) provides a method to reason over resource usage.

We consider r types of renewable resources and a set of activities, denoted by $\mathcal{A} = \{a_1, \dots, a_n\}$. Some activities require b_{ir} instances of resource type r for a fixed processing time p_i. Each resource is available in m_k quantities. At any time, the total number of resources in use cannot exceed the total number of available resources $R(r), \forall r$. The RCPSP feasibility problem, which asks if we can schedule all activities within a given time limit, is NP-complete (Garey and Johnson 1975).

Live Interactive Narrative Scheduling Problem

In this section, we present a formal definition of the Live Interactive Narrative Scheduling Problem (LINSP) and describe its formulation as a mixed-integer linear program, for which fast solvers exist.

Problem Definition

A LINSP problem is defined as a tuple $\langle V, E_{req}, E_{ctg}, U, L, \mathcal{A}, b, R, M\rangle$. Similar to STPU, we rely on a temporal graph G, which contains a set of vertices, V, and two disjoint sets of requirement and contingent edges, E_{req} and E_{ctg}. For each edge $e_{AB} \in E_{req} \cup E_{ctg}$, we define an upper limit U_{AB} and a lower limit L_{AB} on its length. Either limit may be undefined, which we can write as $U_{AB} = \infty$ or $L_{AB} = -\infty$. The limits can also be negative; a negative upper limit U_{AB} implies that v_A must occur after v_B.

To handle RCPSP-like resource constraints, we represent an activity a_i using its starting point v_{S_i} and ending point v_{E_i}. The set of activities $\mathcal{A} = \{a_1 = \langle v_{S_1}, v_{S_1}\rangle, \ldots, a_n = \langle v_{S_n}, v_{E_n}\rangle\}$ where $\forall i, v_{S_i}, v_{E_i} \in V$ and $(v_{S_i}, v_{E_i}) \in E_{req} \cup E_{ctg}$. The length of activity a_i is captured by temporal constraints $U_{S_iE_i}$ and $L_{S_iE_i}$. b_{ir} is the number of type-r resource needed by a_i. Those resources are utilized at time v_{S_i} and released at time v_{E_i}. $R(r)$ denotes the total number of available type-r resource.

Plot choices are represented as sets of time points that are mutually exclusive to each other. We introduce the set of mutual exclusive pairs $M = \{\langle v_A, v_B\rangle, \ldots\}$. Every pair of vertices in M never appear in the same story.

Derived from dynamic controllability of STPU, the feasibility problem of LINSP asks if there is an assignment to the duration of intervals in E_{req} based only on previous observations of E_{ctg} such that all temporal constraints and resource constraints are satisfied. The problem can be extended to support multiple simultaneous interactive narratives.

LINSP Is NP-Hard

It is easy to see that the LINSP can be reduced from RCPSP, so the feasibility problem is NP-hard. Here we provide a sketch of proof.

For every activity a_i in the RCPSP, we create two corresponding time points as its starting (v_{S_i}) and ending (v_{E_i}) points and a requirement edge in-between. The upper and lower limits on the requirement edge are both equal to the duration of the activity p_i. Further, we create a dummy starting point v_0 that precedes every activity and a dummy ending point v_∞ that succeeds every activity. If the RCPSP problem contains any precedence constraints, they are created in the LINSP version accordingly. The derived LINSP problem has no contingent edges or mutual exclusive pairs. The feasibility of RCPSP asks if a viable schedule can be completed in t_{max} time. This can be modeled by a requirement edge from v_0 to v_∞ with a upper limit of t_{max}. The time it takes to create the new LINSP is polynomial to the size of the RCPSP and its solution is equal to that of the original RCPSP. □

A Mixed-integer Linear Program Formulation

We present a formulation of LINSP as a mixed-integer linear program (MiLP). Although MiLP is NP-hard, modern solvers are fast. Cui et al. (2015) present a MiLP formulation for dynamic controllability in STPU. We consolidate their MiLP formulation and a flow-based RCPSP formulation and further present LIN-specific constraints to handle plot choices.

STPU Constraints To reason about temporal uncertainty, we break down the temporal graph $G = \langle V, E_{req} \cup E_{ctg}\rangle$ into a number of triangles, which can be computed by enumerating all possible triangles for small graphs or by finding a minimal triangulation (Heggernes 2006). For any edge, e_{AB}, we introduce two variables representing the its actual upper and lower limit, denoted by u_{AB}, l_{AB}. Afterwards, for each triangle with vertices v_A, v_B and v_C, we introduce a set of constraints for every valid triangle that does not contain mutually exclusive vertices.

We consider one Ctg edge per triangle at a time. If there are two Ctg edges in one triangle, they are considered sequentially with the other treated as a Req edge. Two Ctg edges cannot end on the same time point as two uncontrollable intervals cannot be guaranteed to end at the same time. Thus, triangles with three Ctg edges do not exist.

First, we add shortest path constraints

$$\begin{aligned} &l_{AC} \leq u_{AB} + l_{BC} \leq u_{AC} \\ &l_{AC} \leq l_{AB} + u_{BC} \leq u_{AC} \\ &u_{AC} \leq u_{AB} + u_{BC} \\ &l_{AB} + l_{BC} \leq l_{AC} \end{aligned} \tag{1}$$

If there are no Ctg edges in the triangle, no other constraints are required. Let e_{AC} be the Ctg edge. Then for each triangle, we differentiate among three scenarios: *the follow case*, *the precede case*, and *the unordered case*. In the follow case, $U_{BC} \leq 0$, indicating that v_B is always scheduled after C has been observed, hence no further constraints are necessary.

The precede case happens when $L_{BC} \geq 0$, indicating that B must precede or coincide with C. We would be unaware of the exact $v_A \rightarrow v_C$ duration when scheduling v_B. The following constraints are needed.

$$\begin{aligned} &u_{AB} \leq l_{AC} - l_{BC} \\ &l_{AB} \geq u_{AC} - u_{BC} \end{aligned} \tag{2}$$

The unordered case occurs if $L_{BC} < 0$ and $U_{BC} \geq 0$, indicating v_B can be scheduled independently to v_C. We add the following disjunctive constraint.

$$(l_{BC} < 0) \vee \begin{pmatrix} u_{AB} \leq l_{AC} - l_{BC} \\ l_{AB} \geq u_{AC} - u_{BC} \end{pmatrix} \tag{3}$$

Next, we introduce a waiting period, denoted by wait variable, w_{ABC}, indicating v_B must either wait until after v_C or till at least $U_{AC} - U_{BC}$ duration after v_A.

$$w_{ABC} \geq u_{AC} - u_{BC} \tag{4}$$

For every requirement edge e_{AB} and wait variable w_{ABX} where e_{AX} or e_{BX} is contingent, we have the constraint:

$$l_{AB} \geq min(l_{AX}, w_{ABX}) \tag{5}$$

Different wait variables interact with each other in a process called wait regression. For a wait w_{ABX} and a contingent edge e_{DB},

$$(w_{ABX} < 0) \vee (w_{ADX} \geq w_{ABX} - l_{DB}) \tag{6}$$

If e_{DB} is a requirement edge, we have

$$w_{ADX} \geq w_{ABX} - l_{DB} \tag{7}$$

Due to space restrictions, we refer readers to Cui et al. (2015) and Morris, Muscettola, and Vidal (2001) for a proof of correctness of the formulation above.

RCPSP Constraints We add constraints for handling resource utilization and release using the flow-based continuous time approach (Artigues, Michelon, and Reusser 2003), which is based on the intuition that when an activity terminates, it transfers its resources to other activities that need them and occur later. We create an additional activity a_{n+1} to be a *resource sink* at the end of the narrative for resources not renewed or transferred to the next activity (e.g., props destroyed by the activity).

For each pair of activities $a_i = \langle v_{S_i}, v_{E_i} \rangle, a_j = \langle v_{S_j}, v_{E_j} \rangle$, we introduce a binary variable $x_{i,j}$, which equals 1 if and only if a_i occurs before a_j. Otherwise, it equals 0. The following constraints handle proper sequencing of activities, where M is a very large integer.

$$\begin{aligned} & l_{S_iS_j} \geq u_{S_iE_i} - M * (1 - x_{i,j}), \qquad i < j \\ & x_{i,j} + x_{j,i} \leq 1, \qquad\qquad \forall (i,j) \in \{0, \ldots, n+1\}^2 \\ & x_{i,k} \geq x_{i,j} + x_{j,k} - 1, \ \forall (i,j,k) \in \{0, \ldots, n+1\}^3 \end{aligned} \tag{8}$$

For resource type r, we define a flow variable $f_{i,j,r}$ as the quantity of r flowing from a_i to a_j at the end of a_i. Then $f_{i,j,r}$ value will always be restricted by the amount of r released by a_i and the amount consumed by a_j.

$$\begin{aligned} & f_{i,j,r} \leq min(b_{ir}, b_{jr}) \times x_{i,j}, \\ & \qquad \forall (i,j) \in \{0, \ldots, n\} \times \{0, \ldots, n+1\} \end{aligned} \tag{9}$$

To conserve the total resources, the total r released by event a_i to any subsequent events a_j can never exceed the total resources produced by a_i. Hence, for all r we define

$$\begin{aligned} & f_{n+1,0,r} = R(r) \\ & f_{i,j,r} \geq 0, \qquad\qquad \forall (i,j) \in \{0, \ldots, n+1\}^2 \\ & \sum_{i \in A \cup \{0,n+1\}} f_{i,j,r} = b_{ir}, \ \forall j \in \{0, \ldots, n+1\} \\ & \sum_{j \in A \cup \{0,n+1\}} f_{i,j,r} = b_{jr}, \ \forall i \in \{0, \ldots, n+1\} \end{aligned} \tag{10}$$

Branching Constraints A contribution of this paper is the ability to use plot choices for constraint satisfaction. If a plot choice is controlled by the player, the system must make sure that all choices are feasible. However, the drama manager may also selectively offer plot choices in order to accommodate temporal and resource constraints. See Figure 2(a) for an example story graph, which contains three alternative storylines, which respectively require the player to fight with a sword (v_3 and v_4), to cast magic spells (v_5 and v_6), and to perform a robbery (v_7 and v_8). The drama manager may present some of the storylines to the player depending on how much time the player spends with the wizard ($e_{1,2}$). Note the requirement edge $e_{1,9}$ imposes a limit on the overall length of the story.

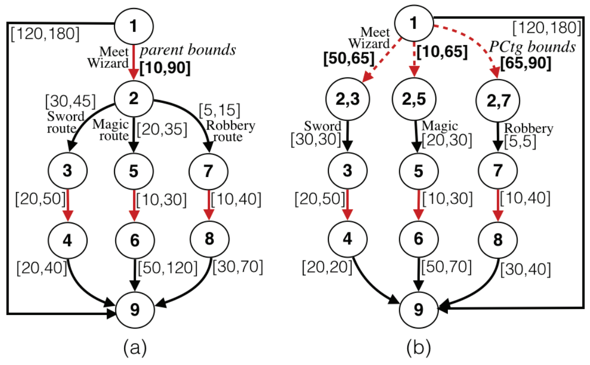
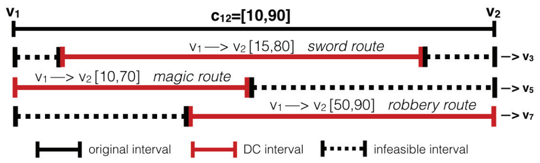


Figure 2: A story with three alternative storylines. Req edges are shown in black. Ctg edges are shown as red solid arrows and PCtg edges are red dashed arrows.



v1
c12=[10,90]
v2
v1 —> v2 [15,80] sword route
—> v3
v1 —> v2 [10,70] magic route
—> v5
v1 —> v2 [50,90] robbery route
—> v7
original interval
DC interval
infeasible interval

Figure 3: The valid ranges of $e_{1,2}$ that are supported by the three alternative storylines respectively. The red durations indicate feasible regions and dashed durations are infeasible regions.

Applying STPU constraints to each alternative storyline individually will lead to the erroneous conclusion that the overall temporal constraint $e_{1,9}$ cannot be met. For instance, the sword-fighting storyline is only feasible when the uncontrollable interval $e_{1,2}$ falls in [15,80]. Figure 3 shows the feasible and infeasible ranges for the three storylines, which individually cannot accommodate the temporal uncertainty induced by $e_{1,2}$, but collectively cover the whole range.

Dynamic controllability suggests that, for every contingent edge that is *shared* by and *precedes* the alternative storylines, we can choose among the storylines based on the outcome of the uncontrollable interval. For example, if the length of the shared interval $e_{1,2}$ is 12, we can present the second storyline to the player, which guarantees the satisfaction of the requirement constraint of $e_{1,9}$. In this section, we introduce additional constraints to allow this behavior.

To capture this intuition, we split the edge $e_{1,2}$ into three separate contingent edges, which we call partial contingent (PCtg) edges, as shown in Figure 2(b). We allow the upper and lower bound of the three edges to change, but require that they collectively cover the original interval $[10, 90]$. We formulate this subproblem as cycle-finding in a directed graph $G^\tau = \langle V^\tau, E^\tau \rangle$. All PCtg edges and the original Ctg

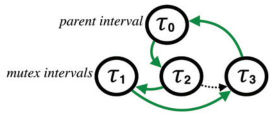


Figure 4: Interval coverage from Fig. 3 represented as finding a cycle in a graph.

edge become vertices in a new graph. With abuse of notation, we use τ_0 to denote the original interval as well the corresponding vertex $\tau_0 \in V^\tau$. The PCtg interval and corresponding vertices are denoted as $\tau_0, \ldots, \tau_m \in E^\tau$. For each interval, we denote its lower limit as l_i^τ and upper limit as u_i^τ. The edges in E^τ are determined by the interval overlaps, which we discuss below. The overall goal of the subproblem is to find a cycle that includes τ_0 and at least one other interval. As an example, Figure 4 contains a valid cycle $\tau_0 \rightarrow \tau_2 \rightarrow \tau_1 \rightarrow \tau_3 \rightarrow \tau_0$. This is similar to cycle-finding problems like the Hamiltonian cycle, but do not require every vertex to be visited.

The edges E^τ are added using the following mechanism. For two split intervals τ_i and τ_j, $i \neq 0$ and $j \neq 0$, if the lower limit of τ_j (denoted as l_τ^j) falls within the interval τ_i, it suggests that concatenating τ_i and τ_j may (albeit not guaranteed to) extend the coverage of τ_i. In this case, we create a directed edge that from τ_i and τ_j. After that, we create an edge from τ_i to τ_0 if the upper limit of τ_0 (denoted as u_τ^0) falls within τ_i, and create an edge from τ_0 to τ_j if the lower limit of τ_0 falls within τ_j.

To encode as MiLP constraints, we create binary decision variables $y_{i,j}$, whose value is 1 if and only if the cycle we found involves the directed edge (τ_i, τ_j). To encode the logical dependence of $y_{i,j}$ on the temporal relationship between τ_i and τ_j, we introduce auxiliary variables z_1^{ij} and z_2^{ij}. $z_1^{ij} = 1$ iff $l_i^\tau \leq l_j^\tau$ and $z_2^{ij} = 1$ iff $l_j^\tau \leq u_i^\tau$.

$$\begin{aligned} (l_i^\tau \leq l_j^\tau) &\vee (z_1^{ij} = 0) \\ (l_j^\tau \leq u_i^\tau) &\vee (z_2^{ij} = 0) \\ y_{i,j} &\leq z_1^{ij} \\ y_{i,j} &\leq z_2^{ij} \end{aligned} \tag{11}$$

The relatioship between the original interval τ_0 and other intervals must also be encoded; the equations are similar and omitted due to space restrictions. We subsequently need the following constraints to enforce the existence of a cycle that go through τ_0.

$$\begin{aligned} &\sum_j y_{i,j} = \sum_{j \in M} y_{j,i}, \forall i \\ &\sum_j y_{j,i} \leq 1, \forall i \\ &\sum_i y_{0,i} = 1 \\ &\sum_i y_{i,0} = 1 \end{aligned} \tag{12}$$

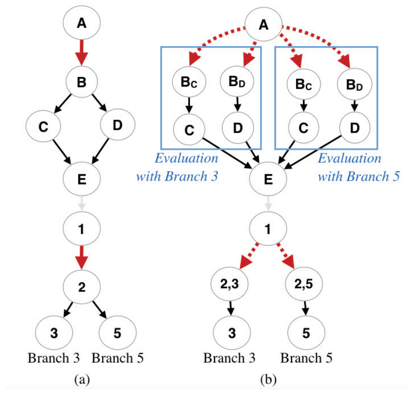


Figure 5: (a) A LIN with two sets of plot choices. (b) The same LIN where the Ctg edges are split into PCtg edges to accommodate additional temporal uncertainty.

Consecutive Plot Choices We discussed the case of one shared contingent edge that precedes one set of plot choices, but there could be multiple contingent edges shared by multiple sets of choices. We can split Ctg edges multiplicatively to cover the entire solution space. Figure 2 shows a graph with two sets of choices. The second Ctg edge, like the earlier case, are split into two PCtg edges. The first edge is split into 4 PCtg edges to accommodate all combinations of plot choices. The same cycle-finding constraints apply for both sets of PCtg edges.

Although this approach finds a feasible solution whenever there is one, the number of PCtg edges grow exponentially with the number of plot choices. However, this is mitigated by two practical factors. First, the total number of plot choices is limited by the amount of narrative materials authored by human designers. It is likely that we do not have a very large number of choices. Second, it is often the case that splitting one or two Ctg edges will create enough flexibility to find feasible solutions for a plot graph. This method lends itself to an interactive design process, where difficult Ctg constraints may be detected using slack variables. More specifically, we can introduce variables ϵ_{ij}^U and ϵ_{ij}^L for a Ctg edge e_{ij} and convert LINSP into an optimization problem that minimizes

$$\sum_{e_{ij} \in E_{ctg}} (U_{ij} - \epsilon_{ij}^U) + (\epsilon_{ij}^L - L_{ij}) \tag{13}$$

where U_{ij} and L_{ij} are the original bounds on e_{ij}. Large values of ϵ_{ij}^U and ϵ_{ij}^L suggest difficult Ctg edges.

Evaluation

Although fast solvers exist, MiLP problems in general are NP-hard. As an evaluation, we test the scalability of the MiLP formulation for LINSP as a function of the number of time points as well as the proportion of uncontrollable intervals in an interactive narrative.

Setup and Results

We aim to simulate typical LIN scenarios by controlling the shapes of the graph and the density of temporal bounds.

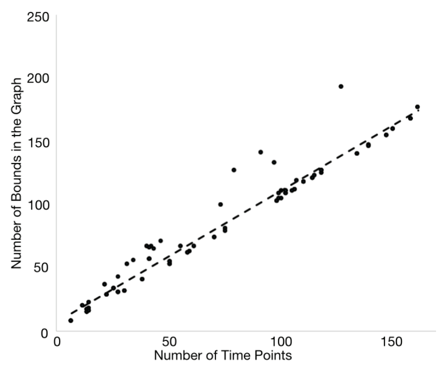


Figure 6: The number of temporal bounds placed on time intervals as the number of time points in the test problems increases.

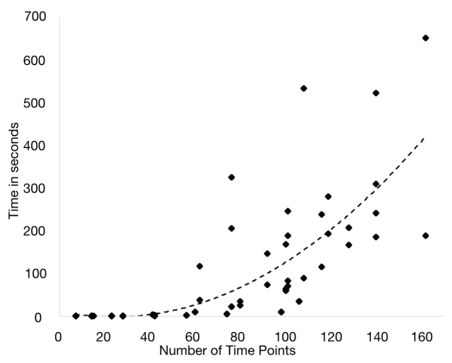


Figure 7: The total time taken to solve LINSP instances as the number of vertices increases.

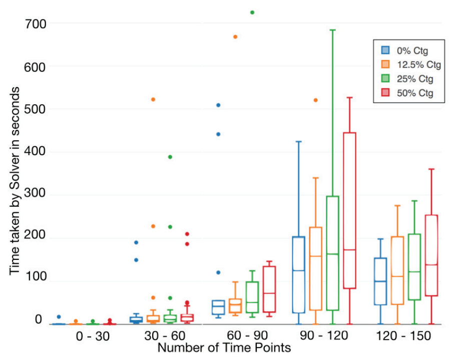


Figure 8: The total time taken to solve LINSP instaces as the number of Ctg edges increases.

Observing plot graphs in choose-your-own-adventure books (Yu 2015, p. 3) and plot graphs learned from crowdsourced stories (Guzdial et al. 2015), we notice that temporal constraints in interactive narratives are mostly local. That is, most edges are between nearby vertices; long-distance edges are rare, though are needed for controlling the total length of a story or a story chapter. For mathematical intuition, consider an ideal graph with n vertices contains only k-vertex components. The number of edges is $O(k^2 \cdot \frac{n}{k})$, which is linear to n. Thus, we created LINSP problems where the number of temporal bounds grows roughly linearly with the number of vertices, as shown in Figure 6.

We created two test suites. The first test suite simulates 77 interactive narratives with temporal and resource constraints. Each test problem contains 2-5 copies of the same interactive narratives running simultaneously, creating a total of 385 problems. The number of vertices in each problem ranges from 15 to 220. We select 50% of the constrained temporal bounds to be resource-consuming activities and place resource constraints on them. Each narrative contains 0 to 72 resources, with up to 3 resources per activity.

The second test suite contains 119 problems, where the proportion of contingent edges are chosen from $\{0\%, 12.5\%, 25\%, 50\%\}$ and the number of vertices ranges from 15 to 150. All problems were solved with the Gurobi solver (Gurobi Optimization Inc. 2016) on a 2.7GHz Intel Xeon E5 12-Core processor with 64GB RAM. Figure 8 shows the results in a box-and-whisker plot. Figure 7 and Figure 8 show the results for two test suits respectively.

Discussion

We observe that most problems in the first test suites can be solved under 400 seconds. As the number of bounds grow linearly with the number of vertices, the time taken to solve LINSP instances grows moderately. The quadratic line fitted to data provides a decent description of the growth trend, with only a few outliers unaccounted for. For the second test suite, we observe a significant increase in the variance in the time needed to solve a problem, as the numbers of vertices and Ctg edges increase. However, the growth still appears to be moderate, as most problems can be solved within 10 minutes. Interestingly, the solution time decreases when going from 90-120 vertices to 120-150 vertices. We postulate that this is because more temporal bounds become redundant and are easily satisfied when the number of vertices increase beyond a point. We conclude the MiLP formulation provides a practical solution for the scheduling of LINs under typical assumptions and with fast solvers, even though the LINSP problem is NP-hard.

Conclusions

In this paper, we present a formulation for event and resource scheduling for live interactive narratives, which combines two existing problems and extends them with additional flexibility to accommodate uncertainty using mutually exclusive plot choices. Numerical evaluation suggests that, under moderate assumptions, the formulation provides practical support for designing and executing interactive narratives in the real world. Future work will address story mediation, or the ability to dynamically change the story (Robertson and Young 2015) while making scheduling decisions. We hope this computational technique will enable novel forms of interactive entertainment that is cost-effective and easy to operate.

References

Arinbjarnar, M.; Barber, H.; and Kudenko, D. 2009. A critical review of interactive drama systems. In *Proceedings of the AISB 2009 AI and Games Symposium*.

Artigues, C.; Demassey, S.; and Nron, E., eds. 2008. *Resource Constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. ISTE-Wiley.

Artigues, C.; Michelon, P.; and Reusser, S. 2003. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research* 149(2):249–267.

Benford, S., and Giannachi, G. 2008. Temporal trajectories in shared interactive narrative. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 73–82.

Brantley, B. 2011. Shakespeare slept here, albeit fitfully. In *The New York Times*. Last retrieved from http://www.nytimes.com/2011/04/14/theater/reviews/sleep-no-more-is-a-macbeth-in-a-hotel-review.html.

Cui, J.; Yu, P.; Fang, C.; Haslum, P.; and Williams, B. C. 2015. Optimising bounds in simple temporal networks with uncertainty under. dynamic controllability constraints. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling*. 52–60.

Fairclough, C., and Cunningham, P. 2003. A multiplayer case based story engine. Technical Report TCD-CS-2003-43.

Garey, M., and Johnson, D. 1975. Complexity results for multiprocessor scheduling under resource constraint. *SIAM Journal on Computing* 4:397–411.

Gurobi Optimization Inc. 2016. Gurobi optimizer reference manual.

Guzdial, M.; Harrison, B.; Li, B.; and Riedl, M. O. 2015. Crowdsourcing open interactive narrative. In *Proceedings of the 10th International Conference on the Foundations of Digital Games*.

Hansen, D.; Bonsignore, E.; Ruppel, M.; Visconti, A.; and Kraus, K. 2013. Designing reusable alternate reality games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Heggernes, P. 2006. Minimal triangulations of graphs: A survey. *Discrete Mathematics* 306(3):297–317. Minimal Separation and Minimal Triangulation.

Hodhod, R., and Magerko, B. 2014. Pharaoh: Conceptual blending of cognitive scripts for computationally creative agents.

Louchart, S., and Aylett, R. 2004. Narrative theory and emergent interactive narrative. *International Journal of Continuing Engineering Education and Lifelong Learning*.

MacVean, A.; Hajarnis, S.; Headrick, B.; Ferguson, A.; Barve, C.; Karnik, D.; and Riedl, M. O. 2011. Wequest: Scalable alternate reality games through end-user content authoring. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*.

Martin, L. J.; Harrison, B.; and Riedl, M. O. 2016. Improvisational computational storytelling in open worlds. In Nack, F., and Gordon, A. S., eds., *Proceedings of the 9th International Conference on Interactive Digital Storytelling*. 73–84.

Morris, P.; Muscettola, N.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, IJCAI'01, 494–499.

Porteous, J.; Teutenberg, J.; Charles, F.; and Cavazza, M. 2011. Controlling narrative time in interactive storytelling. In *The 10th International Conference on Autonomous Agents and Multiagent Systems*, 449–456.

Riedl, M. O., and Bulitko, V. 2013. Interactive narrative: An intelligent systems approach. *AI Magazine* 34(1):67–77.

Riedl, M. O.; Li, B.; Ai, H.; and Ram, A. 2011. Robust and authorable multiplayer interactive narrative experiences. In *The 7th Annual Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Roberts, D. L., and Isbell, C. L. 2008. A survey and qualitative analysis of recent advances in drama management. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 4(2):61–75.

Robertson, J., and Young, R. M. 2015. Interactive narrative intervention alibis through domain revision. In *Proceedings of the 8th Workshop on Intelligent Narrative Technologies*.

Samuel, B.; Ryan, J.; Summerville, A. J.; Mateas, M.; and Wardrip-Fruin, N. 2016. Bad news: An experiment in computationally assisted performance. In *Interactive Storytelling: 9th International Conference on Interactive Digital Storytelling*, 108–120. Springer.

Shilkrot, R.; Montfort, N.; and Maes, P. 2014. nARratives of augmented worlds. In *Proceedings of 2014 IEEE International Symposium on Mixed and Augmented Reality-Media, Art, Social Science, Humanities and Design*.

Stenros, J.; Holopainen, J.; Waern, A.; Montola, M.; and Ollila, E. 2011. Narrative friction in alternate reality games: Design insights from conspiracy for good. In *The DiGRA 2011 Conference*.

Tomai, E. 2012. Towards adaptive quest narrative in shared, persistent virtual worlds. In *Proceedings of the 5th Intelligent Narrative Technologies Workshop*.

Tychsen, A., and Hitchens, M. 2008. Interesting times: Modeling time in multi player and massively multiplayer role playing games. *Leonardo Electronic Almanac*.

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

Winer, D.; Amos-Binks, A. A.; Barot, C.; and Young, R. M. 2016. Good timing for computational models of narrative discourse. In *Proceedings of the 7th Workshop on Computational Models of Narrative*.

Yu, H. 2015. *A Data-Driven Approach for Personalized Drama Management*. Ph.D. Dissertation, Georgia Institute of Technology.