Proceedings of the Fifteenth AAAI Conference on Artificial
Intelligence and Interactive Digital Entertainment (AIIDE-19)

Multi-Agent Narrative Experience Management as Story Graph Pruning

Stephen G. Ware,[1] Edward T. Garcia,[2] Alireza Shirvani,[1] Rachelyn Farrell[1]
Narrative Intelligence Lab
[1]University of Kentucky, [2]University of New Orleans
{sgware, ashirvani}@cs.uky.edu, etgarci1@uno.edu, rfarrell@cs.uky.edy

Abstract

In many intelligent interactive narratives, the player controls an avatar while an experience manager controls non-player characters (NPCs). The space of all stories can be viewed as a story graph, where nodes are states and edges are actions taken by the player, by NPCs, or by both jointly. In this paper, we cast experience management as a story graph pruning problem. We start with the full graph and prune intelligently until each NPC has at most one action in every state. Considering the entire graph allows us to foresee the long-term consequences of every pruning decision on the space of possible stories. By never pruning player actions, we ensure the experience manager can accommodate any choice. When used to control the story of an adventure game, players found our technique generally produced higher agency and more believable NPC behavior than a control.

Introduction

Intelligent interactive narratives in virtual environments have numerous applications in entertainment, training, and therapy. These systems typically invite the player to control one character while an *experience manager* (broadly defined) controls the non-player characters (NPCs) based on the system's aesthetic and pedagogic goals.

Experience management can be viewed as graph traversal. Nodes in a *story graph* (Riedl and Young 2006) represent states of the virtual environment, and edges represent actions that change the state. Actions can be taken by the player, by NPCs, or by both jointly (e.g. a player buys an item from an NPC). In any given state, the player can execute a player action, the experience manager can execute an NPC action, or they can together execute a joint action. Together, player and experience manager choose a path through the graph until they arrive at a terminal state. Deciding which NPC actions to take is challenging for at least two reasons. First, the space of all stories is so large that it is often intractable to explore the entire space, and second, it is difficult to anticipate the player's behavior.

This paper relaxes the first challenge to focus on the second. We treat experience management as a story graph pruning problem. We begin with the full state space graph—every possible state and action—and remove only NPC edges until every NPC has at most one action to perform in any given state, creating a policy that makes it clear what the experience manager should do in any state. We never prune player edges. Our goal is to maintain story quality while allowing the player to take any available action at any time. We describe various pruning techniques and the order in which we apply them. We also present a study demonstrating that this pruning leads to better NPC behaviors than a control. While story graph pruning will be intractable in larger domains, this is an opportunity to discover insights for making decisions when story graphs are too large to generate fully. By having the entire graph in memory when making a pruning decision, we can fully anticipate all the short- and long-term consequences of a decision on the space of possible stories.

Related Work

Story-Graph-Based Systems

Terms like *story graph* and *plot graph* have been used inconsistently in the literature (Thue and Carstensdottir 2018). Our definition follows Riedl and Young's (2006): a graph whose nodes are world states and whose edges are actions (which can be player actions, NPC actions, or joint actions). Story graphs are a common data structure for representing interactive narratives, including non-digital ones like Choose Your Own Adventure books (Swinehart 2009).

Bates (1992) and Weyhrauch (1997) were some of the first to describe experience management as a graph traversal problem jointly solved by the player and an AI experience manager, though their *plot graphs* were defined differently. Weyhrauch used search-based optimization to find ideal paths through plot graphs, while Nelson et al. (2006), Roberts et al. (2007), and Thue and Bulitko (2012) used MDP-based methods to find graph traversal policies. Arinbjarnar, Barber, and Kudenko (2009) survey systems based on graph traversal. While they differ in their graph structures, all frame the problem as joint decision making between the player and the system.

Like many previous systems, we assume our story graph is Markovian—the experience manager makes decisions based only on the current state and does not track the history of how the player arrived at that state. Some systems require story graphs to be acyclic, implying some constraints on history, but we do not require acyclic graphs. The Markov assumption is a limitation for story graph systems, because different actions leading to the same state can suggest very different future actions (Farrell and Ware 2016). Given that our graphs are hundreds of millions of nodes and already straining the limits of what can be feasibly computed, we accept this simplifying assumption for our initial work.

Mediation-Based Systems

Many systems that do not explicitly use story graphs still use them implicitly. Systems that generate narratives at run time are still navigating a story graph; they simply generate it on demand. Dynamic experience managers like these avoid the potentially prohibitive cost of calculating the entire graph in advance but may find it more difficult to reason about the long-term consequences of an action. Kybartas and Bidarra (2016) survey dynamic narrative generation systems.

To cope with large story graphs and unpredictable players, many experience managers form a plan for the narrative based on what they expect the player to do and employ *reactive mediation* when the player deviates from that plan (Riedl, Saretto, and Young 2003). Ideally, the system *accommodates* the player by replanning the story to include the unexpected action (e.g. an important character is killed, so another takes their place). When this is impossible, the system may *intervene* to make the player's action fail (e.g. a gun fails to fire). Intervention subverts the player's mental model of the environment's rules and decreases agency, the player's feeling that they can take meaningful action to affect the story (Wardrip-Fruin et al. 2009). In a graph traversal context, intervention can be viewed as pruning a player action edge from the story graph—that action should have been possible, but the system removed it to prevent the story from being derailed.

Experience managers can also employ *proactive mediation* (Harris and Young 2009). By anticipating the player, the system can avoid intervention by ensuring that every player action can always be accommodated. This work presents a kind of proactive mediation. By considering the entire story graph, we ensure that we never prune player action edges—i.e. we always accommodate and never intervene.

Story Domain

Before describing story graph pruning, we will introduce the story domain from our evaluation, which is used in examples throughout this paper. The domain is inspired by a subset of characters from Ware and Young's (2015) *The Best Laid Plans* and realized in the *Camelot* interactive narrative sandbox tool (Samuel et al. 2018).

The player begins at home, where they learn their grandmother is sick. She gives them a gold coin that can be used to buy medicine. The game features three NPCs. A merchant is in the market selling medicine and a sword. The town guard is in the market watching for criminals. A bandit waits in his camp. The bandit has a coin that he keeps in a chest but is hoping to acquire more items of value such as money and medicine. There are four locations: the player's house, the market, the camp, and a crossroads that connects them all. The game ends when the player returns home carrying the medicine or dies.

Figure 1: The narrative game used in our evaluation. Here, the bandit steals a coin from the player.

Seven actions are available. Characters can walk from one place to another. Characters can take items out of the chest in the bandit's camp. Characters can buy items from the merchant for 1 coin each. If a character is armed, they can steal an item from an unarmed character. One character can attack and kill another, unless the attacker is unarmed and the victim is armed. Characters can loot items from slain characters. Finally, a character who knows the bandit's location can report him to the town guard. Despite its simplicity, this domain yields a surprising number of interesting ways the player can accomplish their goal or die trying.

Intelligent Story Graph Pruning

In this section we define terms relevant to story graphs and the methods we propose for pruning them. Our representation is based on Shirvani, Ware, and Farrell's (2017) formulation of narrative planning with intentionality and belief.

Story Graphs

A *story domain* defines objects and actions. An *object* is a logical constant representing a person, place, thing, or concept. Some objects are *characters*, intelligent agents with beliefs and goals.

An *action* is defined by four things. It has a *precondition*, a conjunction of logical propositions which must be true immediately before it can occur, and an *effect*, a conjunction of logical propositions that become true after the action happens. An action specifies zero, one, or several *consenting characters* who must want to take that action. Finally, an action defines *observing characters* as a function $o(c, s) \in \{true, false\}$ for every character c and state s which defines whether character c observes the action in state s. In short, an action defines when it can happen, what changes when it happens, who is performing the action, and who observes the changes that occur.

Consider, for example, the action where the player uses a coin to buy medicine from the merchant at the market. The player, coin, medicine, merchant, and market are objects. The action's preconditions are that the player and merchant are alive and in the market, the player has the coin, and the merchant has the medicine. The action's effects are that the player has the medicine and the merchant has the coin. The player and merchant are consenting characters—both must have a reason to take that action. The observing characters for the action are any characters in the same location, so if the guard is also in the market, he observes the transaction, and if the bandit is in the camp, he does not observe it.

A *story graph* is composed of nodes representing states and directed edges representing actions. We require no particular commitment to how a state is represented, so long as a *state* completely specifies the configuration of the virtual world, including every character's beliefs about that configuration.

A directed *edge* $n_1 \xrightarrow{a} n_2$ may extend from node n_1 to node n_2 for action a only if the preconditions of a are satisfied in n_1 and taking action a while in state n_1 would change the world state to n_2. A *player edge* is an edge for which the player is a consenting character. An *NPC edge* is an edge for which at least one NPC is a consenting character. An edge can be both a player and an NPC edge (e.g. player buys from merchant); we call these *mixed* edges.

Experience Management

We define a *full story graph* to contain every possible state and every allowable edge. Our goal is to begin with a full story graph and then prune NPC edges until the experience manager has unambiguous directions for what each NPC should do in every state. It is our goal never to prune an edge that requires only the consent of the player, meaning the experience manager will always be able to accommodate any player action. We do allow mixed edges to be pruned, because the NPCs involved may not consent to them.

When actions occur instantaneously, the nodes in an unambiguous story graph would have *either* outgoing player edges *or* exactly one outgoing NPC edge. In other words, in every state, the experience manager would know whether to wait for the player to act or to instruct a specific NPC to act in a specific way. However, our domain is realized in a realtime 3D virtual environment (shown in Figure 1) where actions have an unknown duration.

When actions have duration, we define an *unambiguous story graph* to be one where all nodes may have any number of outgoing player actions but at most one outgoing NPC action per NPC.

When the world transitions to a new state, our experience manager checks if there are any outgoing NPC edges for that node. If so, those NPCs are instructed to begin those actions. When some action (player or NPC) finishes, the experience manager transitions to a new state. If any characters are acting when that transition occurs, their actions are interrupted, unless that same action is also allowed in the new state, in which case the action continues.

Consider, for example, a state where the player (who has a coin) and the bandit (who has a sword) are both at the crossroads. The experience manager must be prepared for the player to take any action, but the bandit should have clear directions to either do nothing or take one action. Say the bandit's directions are to rob the player. The bandit must first walk up to the player, but the player may also be moving around and performing other actions while the bandit approaches. If the player successfully executes an action during that time (e.g. the player walks to the market), the bandit's action is interrupted, and the bandit is given new instructions based on the new state (e.g. follow the player to the market).

Practical Consideration: Belief

Even our small story domain can have an infinite or intractably large full story graph, depending on how one models character beliefs. Many researchers have offered models with trade-offs in realism and efficiency (Bates, Loyall, and Reilly 1992; Porteous, Cavazza, and Charles 2010; Ryan et al. 2015; Eger and Martens 2017; Shirvani, Ware, and Farrell 2017; Shirvani, Farrell, and Ware 2018).

We use an extremely simple model to keep the size of our domain tractable. In addition to propositions describing the physical state of the world, we track 9 special belief propositions: the player's belief about the location of the bandit, the merchant's beliefs about the locations of the two coins, the guard's belief about the location of the bandit, the guard's beliefs about whether the player and merchant are criminals, and the bandit's beliefs about the locations of the player and the three valuable objects (the coins and the medicine).

Pruning

In this section we explain how we prune the story graph for this domain in service of the design goals of our game, which are:

- The game must always be finishable.
- NPCs should act believably.
- The player, not the experience manager, should be responsible for how the story unfolds.

We found the criteria described below to work well in this domain, and we attempt to justify them by explaining our motivations and presenting illustrative anecdotes, but we do not claim they are best for all domains.

The pruning algorithm is simple: for each of the criteria described below (in order), for each state node in the graph (in any order), consider the edges leading out of that state, and prune any edges that meet the criteria. Most criteria are based on the existence of paths in the graph, and since a path is a sequence of action edges, we can think of paths as plans.

Intentionality Pruning Several studies have established that *intentionality*, the tendency of agents to adopt and work toward goals, is an important property of believable character behavior (Riedl and Young 2010; Ware et al. 2014). The first pruning we apply to the story graph is to remove any NPC edges which do not appear intentional. An action is intentional if, for every consenting character, given that character's current beliefs, there exists a sequence of causally-linked actions starting with this action that achieves a goal for that agent and such that every other action in the sequence is also explained. Due to space limitations, we must refer readers to Shirvani, Farrell, and Ware (2018) for full details, but here is a brief description: We say an action edge is intentional when each of its consenting characters believes that edge is a first step on a path that leads to a state where one of that character's goals becomes achieved. After this pruning[1], the story graph contains 388,318,086 nodes and 1,028,110,791 edges. This figure counts only the nodes and edges accessible from the initial state, but to reason about intentionality, the graph also needs to include states that characters believe to be possible but aren't actually possible. These nodes and edges are not included in the count.

Shorter Plan Pruning In a state, if we can find two plans for the same agent to achieve the same goal, we prefer the shorter plan and prune the action that begins the longer plan. These need not be two paths to the same state, only two paths where the same goal has been achieved at the end.

For example, say the guard observes the player kill the merchant. Now the guard wants to kill the player. He could first loot the merchant's sword and then attack the player (a 2 action plan) or he could simply attack with his own sword he is already carrying (a 1 action plan). Thus, the edge where the guard picks up the merchant's sword gets pruned for being the start of a longer plan for the same goal. After this pruning, the graph has 93,608,267 nodes (down 76%) and 248,440,557 edges (down 76%).

Lazy NPC Pruning One design principle of our game is that, given two plans to achieve a goal, we prefer the one with more player actions. Consider the player's goal to buy the medicine. The player could travel to the market, buy the medicine from the merchant, and then return home. Alternatively, the merchant could travel to the player's home and sell them the medicine without requiring the player to leave the house. Though both plans are intentional and equally short, we prefer the former, because it gives the player more opportunity to explore and find their own way to achieve their goals. It also avoids stories in which all NPCs converge on the player at the beginning and then constantly follow the player around, hoping for some specific interaction, such as selling the medicine.

We call this the *Lazy NPC principle*. Given an NPC action explained by some goal (e.g. the merchant traveling to the player's home to sell the medicine), if that NPC expects the player to take some action which can also be explained by the same goal (e.g. the player traveling to the market to buy the medicine), we prune the NPC action. After this pruning, the graph has 58,191,971 nodes (down 38%) and 148,928,950 edges (down 40%).

Unique Ending Pruning Many interactive narratives have several possible endings. Another design principle guiding our experience manager is that the player should be responsible for the ending achieved, not the system. Our experience manager does not prefer any particular ending—that is to say, it is neither working with the player to achieve their goals nor working against the player to thwart them, but rather is trying to provide the ending which is the natural result of the player's choices.

Given two edges for the same NPC, we prune the one which most decreases the number of available endings. This is a *tie-breaking* prune, meaning it will never prune the last edge for an NPC.

Consider that the bandit wants the player's coin, and in general he can get it two ways: by robbing the player or by killing the player and looting the coin. Killing the player limits the number of possible endings to 1 (the player dies), but robbing the player leaves them alive, keeping other endings available.

However, say the player buys a sword from the merchant. Now it is impossible to rob the player, so the bandit's only way to get the coin is to kill the player. This pruning is a tie-breaker, so it will not remove the last edge for the bandit, even if it decreases the number of unique endings. We prioritize acting on one's goals over keeping endings available. Otherwise, the bandit would follow the player everywhere, always one step away from killing the player, but never following through with his plan to attack, which harms the perception of intentionality.

It is important to prune longer plans before unique ending pruning. Consider the longer plan example above, where the guard can attack the player with his own sword (1 action) or pick up the merchant's sword and attack with that (2 actions). Both plans eventually limit the story to one ending, but the first action of the 2-action plan can be taken without limiting endings. It is possible the guard will pick up the merchant's sword, leaving him two ways to complete his goal: attack with his sword or attack with the merchant's. One will be removed by unique ending pruning, but not both, since it is a tie-breaking criteria. If the attack with the merchant sword is removed, the guard will have picked up the merchant sword for no reason. Situations like this are a symptom of assuming the story graph is Markovian. Ideally, once the guard starts one plan he will continue it, but when we do not track the history of actions that brought us to the current state, the only way to know a character's "current plan" is to encode it as part of the state, which would dramatically enlarge this already intractable graph. Eventually, we intend to address this with non-Markovian experience management techniques.

Unique ending pruning targets NPC actions, because the experience manager tries to avoid limiting the player, but it is possible and perhaps even desirable for the player to

[1] In practice, the full story graph for this domain is still too large to generate, so instead of starting with the full graph and pruning based on intentionality, we generate the initial graph using intentionality. In each state, NPCs consider every reasonable plan with 3 or fewer actions, and if they think the plan will achieve one of their goals, the first action is added as an NPC edge from that state. We always include every possible player edge, and do not limit the length of player plans. The result is a story graph equivalent to the graph that would result from pruning the full story graph based on intentionality, assuming no NPC adopts a plan longer than 3 actions. We wish to acknowledge the Louisiana Optical Networking Initiative for providing access to a supercomputer with 1.5 TB of RAM and several days of compute time, which were needed to generate this graph.

limit endings with their actions. When a player limits what endings are available, it can be a notable moment of agency. We plan to investigate these principles in future work.

After unique ending pruning, the graph has 52,262,059 nodes (down 10%) and 138,072,434 edges (down 7%).

Goal Priority Pruning Agents rank their goals from most to least important; in our system, this ordering is author-defined. The guard wants to kill the bandit, but he also wants to be at his post in the market. If the player reports the bandit at the crossroads, the guard will go there, and then he has two options: attack the bandit to fulfill his first goal or return to the market to fulfill his second goal. If he returns to the market, the story graph will have a cycle where the guard constantly walks back and forth between the market and crossroads. Cycles like this are also a symptom of the Markov assumption. We cannot know what the character's current plan is, but this pruning provides a work-around: agents always try to complete their highest priority goal first. Killing the bandit is higher priority, so we prune the action where the guard returns to the market. In the next state, where the bandit is dead, the guard can then act on his lower priority goal of returning to the market. After goal priority pruning, the graph has 30,149,245 nodes (down 42%) and 76,006,520 edges (down 45%).

Cycle Pruning The above prune does not prevent all cycles, so we detect cycles of 3 or more edges and break them. When an NPC has multiple actions they can take in a state, we prune those which are part of a cycle. If every edge in a cycle is that NPC's only action for that state, we prune the one which is part of the longest plan (i.e. we prefer to remove a step that requires two more steps after it to achieve the agent's goal over one that only requires one more step after it). After cycle pruning, the graph has 23,159,543 nodes (down 23%) and 56,783,502 edges (down 25%).

Arbitrary Pruning If, after all of the above, an NPC still has more than one action they could take in a state, we consider all of them equally reasonable and choose one arbitrarily. Also, to save memory, we remove all outgoing edges from terminal nodes, since the game will have ended and no more actions are needed. After this final pruning, the graph has 20,365,197 nodes (down 12%) and 49,669,363 edges (down 13%).

Dead End Pruning The story ends when one of the author's goals is achieved, and it must always be possible for the story to end. We define a *dead end* to be a node from which it is impossible to reach a terminal node. In the final round of pruning, we remove NPC edges to ensure that no dead ends are reachable. Note that we only ever remove NPC edges, never player edges; in other words, we avoid the need to intervene by ensuring the narrative never reaches a state where intervention might be necessary. After dead end pruning, the graph has 20,365,187 nodes (down 5%) and 49,669,351 edges (down 2%).

The existence of reachable dead ends (which were present in the original graph and persist after all the above pruning) demonstrates the need for proactive mediation which considers long-term consequences. Dead ends imply that there exists a path of player decisions that could put the narrative in an unfinishable state. In general, there is no upper bound on the length of such a path, so if an experience manager wants to guarantee that it will never intervene, it may not be enough to look, for example, only one state or only two states ahead.

Evaluation

We claim these pruning techniques achieve our design goals. That the story is always finishable is proved by the absence of dead ends in the final graph (i.e. from every non-terminal state these exists a path to a terminal state). We also claim these pruning techniques result in a high agency experience with believable NPC behavior, and we present the results from a playtest of our game in support.

Experimental Design

We want to compare the experience defined by our story graph to a control. The main phenomenon we want to control for is the human tendency to make narrative sense out of any sequence of events (Bruner 1991). This, combined with genre expectations about adventure games, causes people to attribute intelligence to characters even when they are acting randomly. We want to demonstrate that our techniques produce believable behavior above what people would naturally perceive in this domain no matter what policy the experience manager uses. Therefore, we compare our story graph to one generated randomly. At first glance, this may seem like an easy baseline, but as we will discuss later, most people found even random NPC actions believable; they simply found ours *more believable*.

Like our intelligent story graph, the random story graph allows every possible player action in every state. Additionally, in 75% of states, one NPC action is chosen randomly from all NPC actions possible in that state. The story graph was not generated during play, but offline before play, so all participants experienced the same random story graph.

When we initially tested this story graph, we discovered that NPCs killed the player so frequently that it was almost impossible to achieve the ending where the player returns home with the medicine. We felt this control would be too easily outperformed, so we imposed one further constraint: the simplest plan to achieve that ending (player walks to market; buys medicine; walks home) is guaranteed to be possible. Finally, to ensure it was always possible to finish the game, we perform the same dead end pruning done to the intelligent story graph. The result is a mostly random story graph in which there is at least one way to achieve both endings. It has 21,115,022 nodes and 60,492,852 edges, roughly comparable in size to our pruned graph.

We conducted a study with 20 participants, consisting mostly of Computer Science students at the University of New Orleans. Participants first watched a video explaining the controls of the game and then completed an in-game tutorial in which they became familiar with the game's controls, locations, and characters. In the tutorial, the characters take no actions, but introduce themselves and their goals through dialog when the player interacts with them. We cre-

Table 1: Survey Responses

Statement	Prefer Intelligent	Prefer Random	*p*-value (corrected)	Relative Risk
The characters felt realistic.	16	4	**0.0079**	0.4
The characters reacted to things they saw and ignored things they did not see.	13	7	0.1316	0.7
The characters tried to accomplish their goals.	18	2	**0.0008**	0.2
My actions had a significant effect on the story.	16	4	**0.0079**	0.4

ated this tutorial because, in an earlier version of this experiment, we observed that players significantly preferred whichever version they played first, regardless of treatment, and we attributed this to the novelty of exploring the virtual world. The tutorial ensures that participants have explored the world before playing the game, allowing them to focus on the narrative.

After the tutorial, each participant played two versions of the game: one using the random story graph and the other using the intelligent story graph produced by our pruning. Participants were randomly divided into two groups, with one playing the random version first and the other playing the intelligent version first. Participants were required to complete each version at least twice (to ensure they had a chance to try different strategies), but were invited to play up to ten times. We did not require them to win the game or to experience different endings.

Results

After playing the two versions, participants were shown four statements about character believability and agency and asked to choose whether each was more true of the first version or the second. Table 1 presents the breakdown of results by statement, showing the numbers of participants who preferred the intelligent and the random version.

We hypothesize that players will significantly prefer the intelligent story graph—that is, they will say these statements were more true of the intelligent story graph. A binomial exact test confirmed this hypothesis for three of the four questions at the $p < 0.05$ level. The p-values in Table 1 are given after applying Benjamini and Hochberg's (1995) correction for multiple hypothesis testing. Effect size is given as relative risk that participants preferred the random version.

Due to our relatively small sample size, we did not detect a significant effect at the $p < 0.05$ level for the statement, "The characters reacted to things they saw and ignored things they did not see." However, players still prefer the intelligent version almost 2 to 1 for this statement, so we expect a larger sample would reveal such a trend, and we hope to repeat this study with a larger sample later.

Conclusions and Future Work

In this paper, we frame experience management as a story graph pruning problem. By starting with a full story graph and pruning only NPC actions, we pre-compute the experience manager's policy, accounting for the long-term effects of those decisions on the entire space of possible stories. We ensure NPCs act believably and that the story can always reach an ending while also ensuring the experience manager never needs to prevent the player from taking an action.

We learned several important lessons from this work. First, story graphs, even for small domains, can get very big very fast. Even in our simple domain, when limiting NPC plans to 3 steps and accounting for only 9 beliefs, the graph contains over 300 million state nodes and 1 billion edges, and that number does not count the states which characters believe to be possible but are actually impossible. Pruning a complete story graph will be intractable for most domains, but this work was instructive because it allowed us to consider the long-term consequences of every experience manager decision. We believe these insights can be applied, probably as heuristics, to larger graphs which must be generated on demand.

The second lesson is that, in storytelling domains like this one, random actions are a surprisingly strong baseline. After playing the first version of the game (but before playing the second), participants responded to the four statements in Table 1 on a 5 point Likert scale. Two groups of 10 participants was not a large enough sample for a between subjects analysis, but both groups tended to agree with all four statements, even those who played the random story graph. Anecdotally, several participants invented elaborate explanations to make sense of the random actions they saw and said they enjoyed these "plot twists." Perhaps the human tendency to narrativize events (Bruner 1991) is so strong that most people cannot see actions as random, only as easier or harder to explain, and thus a randomly generated story graph is a stronger baseline than it might seem.

We feel that the most limiting assumption of this initial work is that the story graph is Markovian. Stories are non-Markovian; different action sequences leading to the same state often require different conclusions. In future work, we intend to explore how tracking the history of events can improve experience management and NPC believability.

Artifact: Story Graphs

To make this work reproducible and to encourage others to experiment with story graph pruning criteria, we have released the story graphs described in this paper:

http://cs.uky.edu/∼sgware/projects/storygraphs

Acknowledgments

This research was supported by NSF awards IIS-1464127 and IIS-1647427. Supercomputing was provided by the Louisiana Optical Networking Initiative.

References

Arinbjarnar, M.; Barber, H.; and Kudenko, D. 2009. A critical review of interactive drama systems. In *Proceedings of the AI and Games Symposium at the Adaptive and Emergent Behaviour and Complex Systems Convention*.

Bates, J.; Loyall, A. B.; and Reilly, W. S. 1992. An architecture for action, emotion, and social behavior. In *in Proceedings of the European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 55–68.

Bates, J. 1992. Virtual reality, art, and entertainment. *Presence: Teleoperators & Virtual Environments* 1(1):133–138.

Benjamini, Y., and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)* 57(1):289–300.

Bruner, J. 1991. The narrative construction of reality. *Critical inquiry* 1–21.

Eger, M., and Martens, C. 2017. Practical specification of belief manipulation in games. In *Proceedings of the 13th AAAI International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 30–36.

Farrell, R., and Ware, S. G. 2016. Predicting user choices in interactive narratives using indexter's pairwise event salience hypothesis. In *Proceedings of the 9th International Conference on Interactive Digital Storytelling*, 147–155.

Harris, J., and Young, R. M. 2009. Proactive mediation in plan-based narrative environments. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games* 1(3):233–244.

Kybartas, B., and Bidarra, R. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games* 9(3):239–253.

Nelson, M. J.; Mateas, M.; Roberts, D. L.; and Isbell, C. L. 2006. Declarative optimization-based drama management in interactive fiction. *IEEE Computer Graphics and Applications* 26(3):32–41.

Porteous, J.; Cavazza, M.; and Charles, F. 2010. Applying planning to interactive storytelling: Narrative control using state constraints. *ACM Transactions on Intelligent Systems and Technology* 1(2):1–21.

Riedl, M. O., and Young, R. M. 2006. From linear story generation to branching story graphs. *IEEE Computer Graphics and Applications* 26(3):23–31.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Riedl, M.; Saretto, C. J.; and Young, R. M. 2003. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, 741–748.

Roberts, D. L.; Bhat, S.; Clair, K. S.; and Isbell Jr, C. L. 2007. Authorial idioms for target distributions in TTD-MDPs. In *Proceedings of the 22nd Conference of the Association for the Advancement of Artificial Intelligence*, 852–857.

Ryan, J. O.; Summerville, A.; Mateas, M.; and Wardrip-Fruin, N. 2015. Toward characters who observe, tell, misremember, and lie. *Proceedings of the workshop on Experimental AI in Games*.

Samuel, B.; Reed, A.; Short, E.; Heck, S.; Robison, B.; Wright, L.; Soule, T.; Treanor, M.; McCoy, J.; Sullivan, A.; Shirvani, A.; Garcia, E. T.; Farrell, R.; Ware, S.; and Compton, K. 2018. Playable experiences at AIIDE 2018. In *Proceedings of the 14th AAAI International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 275–280.

Shirvani, A.; Farrell, R.; and Ware, S. G. 2018. Combining intentionality and belief: Revisiting believable character plans. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Shirvani, A.; Ware, S. G.; and Farrell, R. 2017. A possible worlds model of belief for state-space narrative planning. In *Proceedings of the 13th AAAI International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 101–107.

Swinehart, C. 2009. Accessed 22-May-2018.

Thue, D., and Bulitko, V. 2012. Procedural game adaptation: framing experience management as changing an MDP. In *Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment Conference*, 44–50.

Thue, D., and Carstensdottir, E. 2018. Getting to the point: resolving ambiguity in intelligent narrative technologies. In *Proceedings of the joint workshop on Intelligent Narrative Technologies and the Workshop on Intelligent Cinematography and Editing at the 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Wardrip-Fruin, N.; Mateas, M.; Dow, S.; and Sali, S. 2009. Agency reconsidered. In *Proceedings of the Digital Games Research Association*.

Ware, S. G., and Young, R. M. 2015. Intentionality and conflict in The Best Laid Plans interactive narrative virtual environment. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games* 8(4):402–411.

Ware, S. G.; Young, R. M.; Stith, C.; and Wright, P. 2014. The Best Laid Plans.

Weyhrauch, P. W. 1997. *Guiding interactive drama*. Ph.D. Dissertation, Carnegie Mellon University.