Proceedings of the Nineteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2023)

Causal Necessity as a Narrative Planning Step Cost Function

Stephen G. Ware, Lasantha Senanayake, Rachelyn Farrell
Narrative Intelligence Lab
Department of Computer Science, University of Kentucky, Lexington, KY USA
sgware@cs.uky.edu, lasantha.senanayake@uky.edu, rachelyn.farrell@uky.edu

Abstract

Narrative planning generates a sequence of actions which must achieve the author's goal for the story and must be composed only of actions that make sense for the characters who take them. A causally necessary action is one that would make the plan impossible to execute if it were left out. We hypothesize that action sequences which are solutions to narrative planning problems are more likely to feature causally necessary actions than those which are not solutions. In this paper, we show that prioritizing sequences with more causally necessary actions can lead to solutions faster in ten benchmark story planning problems.

Introduction

Interactive narratives used in virtual environments for entertainment and education can be broadly divided into *top-down* and *bottom-up* approaches (Kreminski and Mateas 2021). In bottom-up (or emergent) systems, stories arise from simulation. Top-down systems typically plan their narrative in advance to meet certain goals or structural requirements. This is not to say emergent stories lack structure, only that their structure is a consequence of the simulation rather than constraints that explicitly guide story generation.

Story planning algorithms are frequently used to generate or repair a story during play to guarantee certain structural requirements. Narrative planners have been proposed to ensure narrative milestones are reached (Porteous, Cavazza, and Charles 2010), to ensure characters act according to their beliefs and intentions (Riedl and Young 2010; Teutenberg and Porteous 2015; Ware and Siler 2021), to ensure certain events are salient in memory (Cassell and Young 2013; Farrell, Ware, and Baker 2020), and for a variety of other narrative reasoning tasks. To be used at run time, narrative planners must be fast. Search can often be sped up by reasoning about the structure of the story during planning.

One important structural feature of stories is the network of causal relationships between events. For this paper, we say an earlier event is causally linked to a later event when the earlier event establishes some condition needed by the later event. Further, we say an event is causally necessary if leaving it out of the story would make it impossible for some later event which depended on it to occur.

We hypothesize that prioritizing stories that contain more causally necessary actions—that is, actions which cannot be left out—can guide top-down forward state-space narrative planning algorithms to solutions faster. Using a set of 14 benchmark storytelling domains, 10 of which are the right size to give interesting results, we show that discounting casually necessary actions often performs better than a non-discounted search. We conclude with a discussion of ways our simple method might be improved.

Related Work

Psychological Research on Causality in Narratives

Psychologists have shown that events with more causal connections in a story are judged more important by readers (Trabasso and Sperry 1985) and that the causal network of events is used to organize narratives in memory (Gerrig 1993). When we experience a story one event at a time, whether or not the current event is causally linked to past events is one of several factors that affect how easy it is to remember past events (Zwaan and Radvansky 1998).

The term *causality* can have several meanings. Tapiero et al. (2002) identify four kinds of causality that are important to narrative comprehension: physical causality, motivation, psychological causation, and enablement. One way they distinguish between these types of causality is whether events in a causal relationship are necessary or sufficient to one another. Event A is necessary to event B if, without A, B could not occur. Event A is sufficient to B if B must occur because A occurred. Tapiero et al. suggest that enablement (A enabled B) may be perceived as the weakest causal relationship because A is necessary but not sufficient for B. However, in this paper, we primarily concern ourselves with enablement and necessity because they are easy to detect in plans. Before we proceed, we will briefly discuss the nuances in mapping psychological research to narrative planning.

Many story planners strive to be domain independent, allowing the user to define the story world in predicate logic. The use of arbitrary, user-defined predicates means it is not always straightforward to distinguish the four kinds of causality above in a narrative planning problem. Causal links based on predicates that describe the physical world might be interpreted as physical causality or enablement. Causal links based on predicates describing mental states might be considered motivation or psychological causation. Some narrative planning problems encode motivation as predicates like any other (Kartal, Koenig, and Guy 2014), while some planners reason explicitly about motivation using special modal predicates (Riedl and Young 2010).

Applying psychological research on causality to narrative planners is further muddied by the term *event*. Psychologists tend to use *event* to mean any communicative act in the discourse. For example, in the study by Tapiero et al., this sentence about a rain-soaked book is considered an event: "His soaked manual was almost unreadable." Planning domain authors might consider that a fact or a state rather than an event. An *event* in the context of story planning is usually characterized as changing the world state, so "It began to rain" is an event, whereas "The book was wet" is a fact, the effect of the rain.

In this paper we focus on causally necessary events, where some effect of event A enabled a later event B which otherwise would not have been possible. Further improvements to our method might be achieved with a more precise mapping of different types of narrative causality, though this may come at a cost to domain independence.

Causality in Narrative Generation

Young (1999) proposed AI planning as a way to generate and repair interactive stories, in part because of the rich representation of causality offered by partial order planners. Actions have preconditions which must be satisfied before they can occur and effects which change the world state. Partial order planners maintain an explicit list of causal links—when the effect of an earlier action is used to satisfy the precondition of a later action. The IPOCL (Riedl and Young 2010) narrative planner uses causal links to model intentionality. CPOCL (Ware et al. 2014) uses causal links to model narrative conflict. These planners work backwards from the problem's goals, adding causal links as needed to construct a valid story. While these algorithms highlight the importance of causality, the method we describe in this paper (prioritizing plans with more causally necessary actions) is not directly relevant to this family of algorithms because their actions always form a causal network.

The method we propose in this paper is for forward state-space storytelling algorithms that add actions to the story one at a time starting in the initial state and working toward the end. They perform an expensive backtracking search through the space of possible stories. Examples include Glaive (Ware and Young 2014), IMPRACTical (Teutenberg and Porteous 2015), Sabre (Ware and Siler 2021), HeadSpace (Sanghrajka, Young, and Thorne 2022), and others. Young et al. (2013) provide a survey of narrative planning systems. Our method is also relevant to storytelling systems that use classical planners for storytelling purposes (Cavazza, Charles, and Mead 2002; Porteous, Cavazza, and Charles 2010).

Our method is not directly relevant to scripted behavior systems, like behavior trees (Rabin 2013), and reactive narrative planners, like ABL (Mateas and Stern 2002), Versu (Evans and Short 2013), and Comme il Faut (McCoy et al. 2014). While these systems are often used for forward state-space storytelling, they do not perform the extensive search of all possible stories our method is meant to improve. Rather, they decide at each moment which is the best next action and commit. However, the value of causally necessary actions could be one of many criteria these systems use to choose actions. Marlinspike (Tomaszewski 2011) does this. It is a forward state-space storytelling system that does not perform extensive search but chooses a next scene which best incorporates the player's past actions—i.e. the one with more causal links back to player actions.

Large neural language models also do not use backtracking search, but causal models can improve their story generation abilities. C2PO (Ammanabrolu et al. 2021) performs causal reasoning forward for each action (Why did the character want to take this action?) and backward from each action (What did the character need to take this action?) and tries to build causal links between events where *wants* and *needs* overlap. Similarly, the Neural Story Planner (Ye et al. 2022) follows the same causal means-end reasoning as a partial order planner to create a causal network of events, but using a language model rather than a symbolic planning domain.

Method

We hypothesize that action sequences which contain more causally necessary actions are more likely to be solutions to narrative planning problems than sequences with fewer causally necessary actions. For simplicity, we define a causally necessary action to be one which, if left out of a sequence, would make that sequence impossible to execute. We test this hypothesis by defining a cost function which discounts actions that are causally necessary and testing its performance as a guide for search in a handful of narrative planning benchmark problems.

Forward State-Space Planning

We define our method in the context of forward state-space planning. Let the state space of a problem be a directed graph whose nodes are states. An edge $s_1 \xrightarrow{a} s_2$ exists from state node s_1 to state node s_2 and is labeled with action a if and only if action a can be taken in state s_1 and doing so would change the state to s_2.

Search begins in the initial state s_0 and follows edges forward until it reaches an acceptable terminal state. A classical planner simply defines a goal proposition; any state where that proposition holds is an acceptable terminal state, and any path from s_0 to a terminal state is a valid plan (Russell and Norvig 2009).

Narrative planners typically place additional constraints on which paths are considered valid solutions. Requirements differ by planner, but they typically define some model of believable character behavior and require that the path to the problem's goal be composed only of believable actions. We use the Sabre narrative planner (Ware and Siler 2021) in our experiments. Sabre's model of believable character behavior reasons about intentionality and character beliefs. When a character takes an action, the action is believable if and only if it can be the first action in a plan the character believes will improve their utility.

Our definition of causal necessity does not rely on Sabre's particular model of believability, so for details we refer readers to the full description of that planner (Ware and Siler 2021). Though we tested our method in Sabre, our method can apply to non-narrative forward planners used for storytelling (Porteous, Cavazza, and Charles 2010) and other forward narrative planners like IMPRACTical (Teutenberg and Porteous 2015) and HeadSpace (Sanghrajka, Young, and Thorne 2022) which use different definitions of believability.

Causal Necessity

Let π be a sequence of n actions $\{a_1, a_2, ..., a_n\}$. Let $\alpha(\pi, s)$ denote the state of the world if we begin in state s and then take each action in π in that order. $\alpha(\pi, s_0)$ is defined if and only if, for all $0 < i \leq n$, the edge $s_{i-1} \xrightarrow{a_i} s_i$ exists in the state space. In other words, $\alpha(\pi, s_0)$ is defined if π is a valid path through the state space that starts at the initial state.

We define a causally necessary action to be one which, if left out, would make the plan impossible to execute. Formally, for some sequence $\{a_1, a_2, ..., a_n\}$, action a_i (where $i < n$) is causally necessary if $\alpha(\{a_1, ..., a_{i-1}, a_{i+1}, ..., a_n\}, s_0)$ is undefined. In other words, if it is left out of the sequence, the actions that remain no longer form a valid path. Otherwise, we say the action is causally unnecessary (the sequence without the action is still a valid path).

There is one special case to consider: the last action in the sequence. We might say the last action is always unnecessary, since it can always be left off. We might also say the last action is always necessary by virtue of being the last action. Instead, we apply a special test for the last action. We define action a_n to be necessary if it achieves a goal. For Sabre, this means that the utility in state s_n is higher than utility in the previous state s_{n-1}. We write this as $u(s_{n-1}) < u(s_n)$.

Causal Necessity Cost

Now we can define a cost function which discounts causally necessary actions.

In classical planning, all actions are assumed to have a unit cost. That is, the cost of a sequence of actions $\pi = \{a_1, a_2, ..., a_n\}$ is $|\pi|$ or n.

For our cost function, causally unnecessary actions still cost 1, but causally necessary actions cost a constant value ϵ, where $0 < \epsilon < 1$. We define our cost function as follows:

$$cost(\pi) = \sum_{i=1}^{|\pi|} \begin{cases} 1 & \text{if } i = |\pi| \text{ and } u(s_{i-1}) \geq u(s_i) \\ \epsilon & \text{if } i = |\pi| \text{ and } u(s_{i-1}) < u(s_i) \\ 1 & \text{if } \alpha(\{a_1, ..., a_{i-1}, a_{i+1}, ..., a_n\}, s_0) \text{ defined} \\ \epsilon & \text{otherwise} \end{cases}$$

The first two cases above are for the last action a_n. When a_n does not improve utility, it is unnecessary and costs 1; otherwise, it is necessary and costs ϵ. The last two cases are for all other actions a_i where $i < |\pi|$. Causally unnecessary actions cost 1, and causally necessary actions cost ϵ.

In our experiments, detailed later, we found $\epsilon = 0.4$ to perform well across a variety of problems, so we will use that value in the following examples.

Motivation

Consider the example interactive narrative planning problem introduced with Sabre. The player starts in their cottage with one coin and a quest to get a potion and return home. There is a market, where a merchant who wants money is selling the potion or a sword for one coin each. An armed guard is stationed at the market to punish criminals. An armed bandit lurks in a nearby camp and schemes to get valuable items like the coin and potion. A crossroads connects the cottage, market, and camp. Characters can walk between locations, give an item to another character, buy an item from the merchant, attack another character if they are armed, and take items from slain characters. The story ends when the player completes their quest by returning home with the potion or dies.

Our hypothesis about the value of causally necessary actions arose when we observed a weakness in forward state-space story planners. Several actions have satisfied preconditions when the story above begins: all four characters can walk to the crossroads, the guard can attack the merchant, or the merchant can give either item away to the guard. Some of these actions are not believable—namely the guard leaving the market, the guard attacking the merchant, and the merchant giving items away—but they are possible, so Sabre will explore them as possible first actions before ultimately rejecting them.

Suppose the first event in the story is that the player walks to the crossroads. The state changes, but the list of actions with satisfied preconditions remains about the same. The options for where the player can walk have changed, but otherwise the same set of actions remains possible: other characters can walk to the crossroads, the guard can attack the merchant, and the merchant can give items away to the guard. The planner's options for the second action in the story are very similar to its options for the first action.

A human storyteller looking at this problem would probably choose a second action that builds on the first; otherwise, what was the point of the first action? A planner, however, has no such intuition, and it would consider any of the available second actions an equally good choice. Each choice represents a new branch in the search, quickly leading to the combinatorial explosion that prevents narrative planning from scaling to large problems. By incentivizing action sequences with more causally necessary actions, we hope to encourage the planner to reuse its past actions.

Figure 1 shows some example calculations for two story branches in this domain. Note that an action which is unnecessary in Story A2 (the player walking to the market) becomes necessary after another action is added (the player buying the potion) in Story A3. Also note that the total cost of a story is not monotonically increasing. Story A4's total

Story 1	**Cost**
The player walks to the crossroads.	1.0
Total	1.0

Story A2	**Cost**
The player walks to the crossroads.	0.4
The player walks to the market.	1.0
Total	1.4

Story A3	**Cost**
The player walks to the crossroads.	0.4
The player walks to the market.	0.4
The player buys the potion from the merchant.	1.0
Total	1.8

Story A4	**Cost**
The player walks to the crossroads.	0.4
The player walks to the market.	0.4
The player buys the potion from the merchant.	1.0
The player walks to the crossroads.	1.0
Total	2.8

Story A5	**Cost**
The player walks to the crossroads.	0.4
The player walks to the market.	0.4
The player buys the potion from the merchant.	1.0
The player walks to the crossroads.	0.4
The player walks home.	0.4
Total	2.6

Story B2	**Cost**
The player walks to the crossroads.	1.0
The bandit walks to the crossroads.	1.0
Total	2.0

Story B3	**Cost**
The player walks to the crossroads.	0.4
The bandit walks to the crossroads.	0.4
The bandit slays the player.	0.4
Total	1.2

Figure 1: Some example cost calculations where $\epsilon = 0.4$. Branch A shows a story that achieves one acceptable terminal state where the player completes the quest. Branch B shows a story for the other acceptable ending where the player dies.

cost is higher than Story A5, even though Story A5 is a longer version of Story A4. The same is true of B3 and B2.

Limitations

There are at least two significant limitations to our cost function. First, actions which will eventually become necessary may have a high cost until they do. When an action has two or more causal parents (that is, it directly relies on two previous actions) those causal parents might be unnecessary until their causal child is added. This is illustrated by the B story in Figure 1. The bandit cannot slay the player until they are both at the crossroads, but the slay action depends on two previous and otherwise unnecessary actions.

Second, our cost function only measures whether an action is necessary for the story to be a valid sequence of actions; it is not measuring *narrative necessity*. We see this in Story A5. Buying the potion is necessary to achieve the player's quest, but if it is left out of the story, the remaining actions are all still possible to execute. Another example can be seen in the *Treasure Island* domain introduced by Shirvani et al. (2017), where the first action in the solution is for Jim Hawkins to spread a rumor that there is treasure on the island. This action does not change the physical state of the world, only the beliefs of Long John Silver. The rumor action is not causally necessary according to our definition, because without it the remaining actions in the story can still be executed. However, the rumor is narratively necessary because without it Long John Silver has no motivation to seek the treasure.

This raises an obvious question: Why not modify the cost function to detect narratively necessary actions? For Sabre at least, this would be expensive. Checking whether actions would still make sense if an earlier action is left out (i.e. Would Long John Silver still be willing to sail to the island if he does not believe there is treasure on it?) would require additional costly searches that would make our cost function expensive to calculate in many cases. We have intentionally kept this cost function simple in the hopes that it will be efficient to calculate and can potentially be applied in other systems. We acknowledge that causal necessity is not narrative necessity, and this is a limitation of our method.

Evaluation

We tested the effectiveness of discounting causally necessary actions for three kinds of simple forward state-space search. All three algorithms begin in the initial state s_0 and add actions to the end of the plan π until π is a solution according to Sabre. We tested all three algorithms with ten different values of ϵ in $\{0.1, 0.2, ..., 0.8, 0.9, 1.0\}$.

- *Uniform Cost Search* (UCS) always extends the plan π which has the lowest $cost(\pi)$, where $cost$ is the function we defined earlier. Note that when $\epsilon = 1$ this search is identical to *Breadth First Search*, since all actions cost 1 regardless of whether they are causally necessary or not.
- *A* Search* always extends the plan π which has the lowest $cost(\pi) + h(\pi)$, where $cost$ is the function we defined earlier and h is a heuristic that estimates the number of actions remaining before the plan is a solution. We use Bonet and Geffner's additive heuristic (2001). Note that when $\epsilon = 1$ this search is a default $A*$ search where all actions have unit cost.
- *Explanation First Search* (EFS) is a variant of $A*$ that first performs the search needed to discover if an action is explained before it performs the search to discover if the action can lead to the problem's goal (Siler and Ware 2022). Among all plans π which contain only explained actions, it extends the π with lowest $cost(\pi) + h(\pi)$.

Domain	# Characters	# Fluents	# Actions	# Triggers	Max $\|\pi\|$	Max Exp.	Max Epi.	Best Search
Bribery	3	25	27	0	5	5	2	too small
Deer Hunter	3	42	28	76	10	5	1	A* $\epsilon = 0.1$
Secret Agent	2	15	44	75	8	8	1	EFS $\epsilon = 0.1$
Aladdin	5	150	282	378	11	8	3	too large
Hospital	4	67	102	196	10	6	3	A* $\epsilon = 0.5$
Basketball	4	93	168	192	8	5	3	too large
Space	2	26	29	66	5	3	2	EFS $\epsilon = 0.6$
Fantasy	4	85	76	141	9	3	3	A* $\epsilon = 0.3$
Western	4	99	352	637	8	5	2	EFS $\epsilon = 0.3$
Raiders	3	29	35	66	7	4	2	EFS $\epsilon = 0.2$
Gramma	4	96	800	952	5	4	2	EFS $\epsilon = 1.0$
Treasure	2	9	5	0	4	4	3	too small
Jailbreak	3	46	106	54	7	7	2	A* $\epsilon = 1.0$
Lovers	3	40	312	375	5	5	2	EFS $\epsilon = 1.0$

Table 1: Summary facts for benchmark problems. After Sabre grounds and simplifies each domain, this table gives the number of agents (characters), number of fluents which describe a state, number of actions, and number of triggers in the domain. It also shows the limit on the length of a story (Max $|\pi|$), limit on the length of an agent's explanation for their actions (Max Exp.), and limit of epistemic nesting (Max Epi.). For problems that gave interesting results, it gives the best performing search and its ϵ value. If several ϵ values tied for best, we give the lowest.

We did not test $\epsilon = 0$, since this would make the searches incomplete. We can quickly prove this by example. It is possible to construct an infinitely long sequence of causally necessary actions. In our example domain, if the player simply walks back and forth between the cottage and crossroads, all but the last action will be causally necessary, resulting in an arbitrary number of sequences that would all cost 1 when $\epsilon = 0$.

We ran all three of these search techniques, for all ten values of ϵ, on each of the benchmark problems described below. We ran each test 10 times, shuffling the order of actions[1] in the domain each time, to measure the average number of state space nodes visited during a search. Each search was allowed to visit up to 1,000,000 nodes (that is, to consider 1 million possible stories) before ending in failure. All searches were performed on a Dell Precision 5820 desktop computer with a 4.10GHz Intel Xeon W-2225 CPU and 512 GB of RAM.

Benchmark Problems

We gathered benchmark narrative planning domains from the literature to test these search methods. Most were originally described in the Planning Domain Definition Language (PDDL), which distinguishes between a planning domain—its types and actions—and a planning problem—its initial state and goal. For most of the domains below, each domain is associated with a single problem, so we use "domain" and "problem" interchangeably. For domains with more than one associated problem, we chose a representative small example problem.

Sabre reasons about two phenomena beyond classical planning: character intentions and character beliefs. Some domains were not originally designed with one or both of these in mind, so we have made an effort to add them. For problems without explicit intentionality, it was straightforward to determine which characters were the consenting characters based on when characters appear in an action's parameters and the semantics of example stories from that domain. For example, the $steal$ action in the *Basketball* domain clearly specifies which character is stealing the item (and is thus a consenting character) and which is an unwitting victim (not consenting). For problems without explicit beliefs, we used simple conventions. For example, characters observe actions if they happen at their location and do not observe actions at other locations.

- The appendix of Riedl's dissertation (2004) provides four problems originally used by the IPOCL planner: *Bribery*, *Deer Hunter*, *Secret Agent*, and *Aladdin*. They include explicit intentionality, but we added beliefs. Of these, *Bribery* was too small to show significant differences between search techniques, and *Aladdin* was too large for any search technique to solve before visiting 1 million nodes.
- Porteous has released[2] two planning domains used in storytelling systems. *NetworkING* (Porteous, Charles, and Cavazza 2013) generates medical drama stories based on the relationships between characters. Even a small example problem in this domain was too large for any of our search techniques. A prototype system by Kartal et al. (2014) used Monte Carlo Tree Search in a planning domain to generate crime stories. We refer

[1] Sabre builds a tree data structure similar to Fast Downward's (Helmert 2006a) to efficiently detect which actions have satisfied preconditions in a given state. This tree dictates the order in which actions are tried during search, but when ties arise in the tree construction process, the original action order can affect search performance. We shuffle the action order to control for any performance that arises from lucky action orderings.

[2] Julie Porteous's website: https://porteousjulie.bitbucket.io/

Domain	Baseline $\epsilon = 1.0$				Best Search					% Improvement	
	Visited	Visited σ	Time (ms.)	Time σ	ϵ	Visited	Visited σ	Time (ms.)	Time σ	Visited	Time
Uniform Cost Search											
Deer Hunter	10885	13	788	58	0.4	1809	23	164	44	83%	79%
Secret Agent	179	1	20	13	0.1	145	2	15	4	19%	25%
Hospital	-	-	-	-	-	-	-	-	-	-	-
Space	4503	343	310	23	0.6	2416	44	148	24	46%	52%
Fantasy	-	-	-	-	0.3	235543	2379	25014	591	-	-
Western	-	-	-	-	-	-	-	-	-	-	-
Raiders	12681	2930	677	151	0.4	7400	327	430	31	42%	36%
Gramma	24027	117	8329	501	1.0	24027	117	8329	501	0%	0%
Jailbreak	-	-	-	-	-	-	-	-	-	-	-
Lovers	-	-	-	-	-	-	-	-	-	-	-
A* Search											
Deer Hunter	68	0	30	5	0.1	42	0	40	11.12	38%	-33%
Secret Agent	38	0	12	2	0.1	34	0	14	5	11%	-20%
Hospital	-	-	-	-	0.5	118874	89	136715	5736	-	-
Space	2838	32	350	22	0.6	1447	13	146	11	49%	58%
Fantasy	-	-	-	-	0.3	233927	2350	49328	1297	-	-
Western	813322	62888	789892	80072	0.7	788655	95742	751735	96538	3%	5%
Raiders	1382	285	128	29	0.2	1201	555	108	46	13%	16%
Gramma	8596	156	18955	1153	1.0	8596	156	18955	1153	0%	0%
Jailbreak	399402	0.0	96131	4063	1.0	399402	0	96131	4063	0%	0%
Lovers	-	-	-	-	-	-	-	-	-	-	-
Explanation First Search											
Deer Hunter	93	0	22	3	0.1	72	0	18	3	23%	19%
Secret Agent	32	0	12	2	0.1	32	0	12	2	0%	0%
Hospital	-	-	-	-	-	-	-	-	-	-	-
Space	652	21	115	6	0.6	358	4	54	8	45%	53%
Fantasy	85732	0	27006	1245	0.2	23952	17	6822	102	72%	75%
Western	-	-	-	-	0.3	349154	1403	427302	36809	-	-
Raiders	386	0	44	4	0.2	340	3	35	4	12%	21%
Gramma	3805	172	3655	344	1.0	3805	172	3655	344	0%	0%
Jailbreak	-	-	-	-	-	-	-	-	-	-	-
Lovers	108893	186	23519	1977	1.0	108893	186	23519	1977	0%	0%

Table 2: Results for each search technique on each problem. This table gives the number of nodes visited and time spent to search for a solution for the baseline version of each search technique ($\epsilon = 1.0$) and for the best performing version of each search technique, along with the ϵ value that gives that best performance. Percent improvement over the baseline is also given. Values are an average of 10 runs and include standard deviations as σ. If several ϵ values tied for best, we give the lowest.

to this domain as *Basketball*. We added intentions and beliefs. We adapted a small example problem with three citizens, who can commit two kinds of crimes and play basketball to relieve anger, and one detective, who can gather clues and arrest criminals.

- The appendix of Ware's dissertation (2014) provides five problems originally used by the Glaive narrative planner: *Space*, *Fantasy*, *Western*, *Raiders*, and *Best Laid Plans*. They include explicit intentionality, but we added beliefs. The last of these domains was used in *The Best Laid Plans* (Ware and Young 2015), and a smaller version of it that incorporated beliefs was later released in the *Save Gramma!* interactive narrative (Ware et al. 2022). We use this version, which is the example domain described earlier, and refer to it as *Gramma*.
- Shirvani et al. (2017) describe a domain with explicit intentionality and beliefs based on *Treasure Island*, but this domain was too small to show significant differences between search techniques.
- Farrell and Ware (2020) provide a domain with intentions and beliefs for manipulating the salience of events in prison escape stories. We call this problem *Jailbreak*.
- Farrell and Ware (2020) provide a randomizable domain with intentions and beliefs for performing belief and intention recognition. We call this domain *Lovers*. We used a version of the problem that requires one agent to lie in order to reach a solution.

Table 1 gives summary facts about each domain. It includes the numbers of characters and actions in a problem. PDDL allows only Boolean predicates when defining a state, but Sabre uses multi-valued fluents—essentially variables which can have one of several possible values (Helmert 2006a). Sabre also allows problems to define triggers, which have preconditions and effects similar to actions, except that they must happen when their preconditions are satisfied. Triggers are similar to but not identical to PDDL axioms, and they are typically used for belief updates (e.g. when one character is in a room with another character, the characters realize that each other are there). Before planning, Sabre grounds and simplifies the problem to remove fluents whose values can never change and actions or triggers which can never occur. The numbers in Table 1 reflect the size of the problem after grounding and simplification.

Sabre performs best when upper limits are placed on the size of a plan, the length of an explanation used to justify a character action, and the depth of epistemic nesting (what character x believes character y believes, etc.). We chose limits based on the known solutions to these problems. For example, the shortest solution to *Gramma*, where the player dies, has only 3 actions, but the shortest solution for the best ending, where the player succeeds on their quest, has 5 actions, so we chose 5 as the upper limit on plan length.

Results

After ruling out problems that were too small or too large, we were left with 10 that gave interesting results. Results for Uniform Cost Search (UCS), A* Search, and Explanation First Search (EFS) on each of these problems is given in Table 2. For each of these three search techniques, we use the version where $\epsilon = 1.0$ as the baseline. UCS when $\epsilon = 1.0$ is equivalent to Breadth First Search; A* when $\epsilon = 1.0$ is equivalent to generic A*, etc.

Standard deviations for the number of nodes visited and time spent during search were low enough across 10 runs to suggest that the values reported are representative for these problems. For example, the UCS baseline takes about 13 minutes to solve *Western*, with a standard deviation of 1.3 minutes. The search visited 813322 nodes on average, with a standard deviation of 62888 nodes, or about 8%.

Discounting causally necessary actions usually improves performance. For example, when we do UCS on the *Deer Hunter* problem with $\epsilon = 0.4$, the search visits 1809 nodes (on average), which is 9076 fewer nodes than the baseline UCS search where $\epsilon = 1.0$, which visits 10885 nodes, an improvement of 83%.

Some problems which could not be solved using the baseline method can be solved by discounting causally necessary actions. For example, baseline UCS cannot solve the *Fantasy* problem in under 1 million nodes, but discounting causally necessary actions with $\epsilon = 0.3$ allows it to be solved. This also happens for *Hospital* and *Fantasy* with A*, and for *Western* with EFS.

We observed some negative results as well. Discounting causally necessary actions was not helpful for the *Gramma*, *Jailbreak*, and *Lovers* problems in any search method (i.e. the baseline $\epsilon = 1.0$ performed best on these problems). However, for *Lovers*, ϵ had very little influence on any of the search methods. For EFS on *Secret Agent*, ϵ had no effect (i.e. all values of ϵ performed the same).

Figure 2 shows the effect of different values of ϵ on each search method for each problem. A* and EFS generally outperformed UCS, though neither A* nor EFS was consistently the best. Lower values of ϵ tended to be better. One notable exception is *Western*, where lower values of ϵ were better for EFS but higher values were better for A*. For *Raiders*, values between 0.3 and 0.7 were best. For *Gramma*, higher values of ϵ were better, though no value of ϵ beat the baseline. These three domains all contain examples of causally necessary actions which must be added early to a plan but do not become necessary until later. *Western* is the most extreme example; the first action of the shortest solution requires one of the characters to get bitten by a snake, and the last action has that character die of the snakebite, with the other actions in between being an unsuccessful attempt to save that character's life. We suspect that solutions that have long distances between when a necessary action is added and when it becomes necessary will benefit from higher values of ϵ.

Different values of ϵ often had an effect on UCS, but ϵ generally had a smaller effect on A* and EFS. We suspect this is because the differences in the heuristic estimates are large enough to eclipse our cost function. Like most planning heuristics, the additive heuristic we used for A* is not admissible and can overestimate on some problems, sometimes dramatically. This may suggest that while discounting causally necessary actions is helpful, a better heuristic has a larger impact on the search. The effect of discounting relative to the heuristic may become more pronounced by using Weighted A*.

Overhead

We also need to verify that our cost function is not too expensive to compute. For each action in a sequence, we need to check whether the sequence can still be executed if we leave that action out. This costs $O(n^2)$, but considering that planning is P-SPACE complete (Helmert 2006b), and that narrative planning is still limited to small problems, we expect this to be negligible relative to the overall cost.

To test this empirically, we considered the *Space* problem, since it can be solved by all search techniques and has a relatively long solution of 5 actions. A longer solution is important because the cost of checking which actions can be left out of a sequence increases as the length of the sequence increases.

We compared Breadth First Search (BFS) to UCS with $\epsilon = 1.0$. These searches will visit exactly the same nodes in exactly the same order, except that UCS will incur the additional overhead of calculating the cost function. Similarly, we compared generic A* to A* using our cost function with $\epsilon = 1.0$. We ran each search 100 times without shuffling action order.

BFS and UCS both visited 4256 nodes every time. On average, BFS took 263.11 milliseconds and UCS took 262.42 milliseconds. UCS actually took about 200 nanoseconds less per node visited, however this extremely small amount of time is more likely due to small variations in the Java garbage collector or the underlying operating system than differences in the search.

Generic A* and A* with $\epsilon = 1.0$ both visited 2777 nodes every time. On average, A* took 301.30 milliseconds and A* with $\epsilon = 1.0$ took 301.54 milliseconds. A* with the cost overhead took about 100 nanoseconds more per node visited than A* without the cost overhead, but again this is a very small difference.

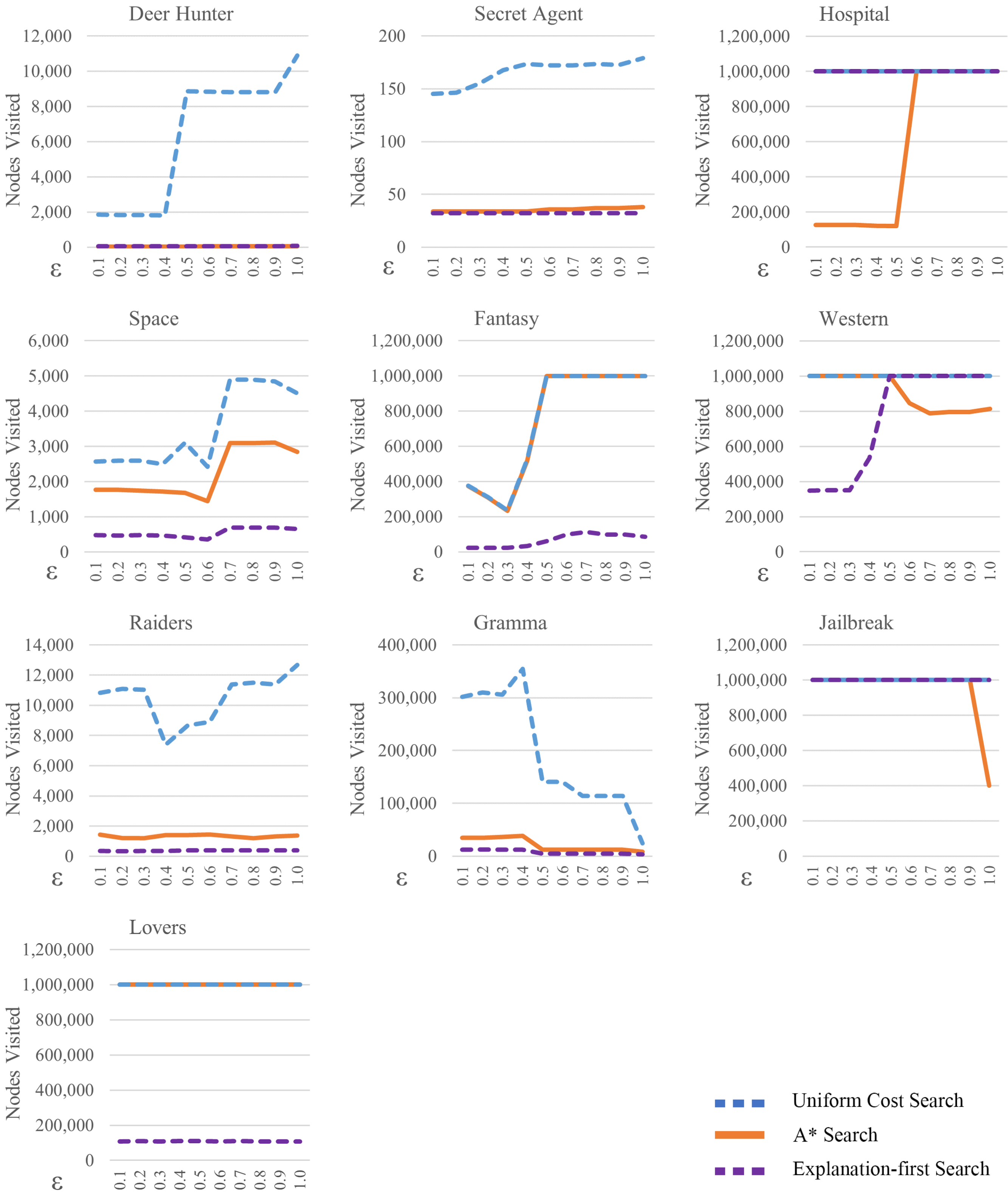

Figure 2: These graphs plot values of ϵ on the x axis and the number of nodes visited by UCS, A*, and EFS on the y axis for each problem. Search is faster when it visits fewer nodes, so lower values on the y axis are better. The upper limit on search was 1 million nodes; a value of 1 million on the y axis indicates the search failed.

We conclude that our cost function is negligible to calculate relative to the other costs of planning. For example, the costs of calculating the heuristic for A* adds about 40,000 nanoseconds per node visited for this problem.

Conclusion and Future Work

The causal network of how the effects of earlier actions enable the preconditions of later actions in a story has been thoroughly studied by psychologists and shown to influence how we perceive stories (Trabasso and Sperry 1985; Zwaan and Radvansky 1998; Tapiero, van den Broek, and Quintana 2002). In this paper, we proposed a simple, low-overhead method to detect causally necessary actions by testing whether or not they can be left out of an action sequence. We hypothesize that discounting causally necessary actions will allow a forward state-space narrative planner to find solutions faster. We provided preliminary support for that hypothesis on a collection of benchmark narrative planning problems. However, the best search technique and best discount factor, and whether large or small discounts were better, varied by problem.

There are many clear directions for future work. We would like to gather more benchmark problems and repeat these tests with a higher limit on the number of nodes visited to demonstrate greater generalization. Weighted A* may be more effective than basic A* at allowing causal necessity discounts to influence heuristic-guided search. We would like to test additional heuristics, like Fast Forward (Hoffmann and Nebel 2001) and the Causal Graph Heuristic (Helmert 2006a), as well as heuristics that account for narrative structure, like the Glaive Heuristic (Ware and Young 2014).

We kept our method simple in the hopes that it could be easily implemented and efficiently calculated in a variety of domain-independent forward planners, including both narrative planners and off-the-shelf classical planners. We focused on necessity and enablement, but Tapiero et al. (2002) identified several kinds of narrative causality with differing effects on perception. Narrative planners may be able to leverage more nuanced concepts of causality if they can distinguish between physical causality, motivation, psychological causation, and enablement. Earlier, we discussed the difference between causal necessity (necessary for the execution of actions) and narrative necessity (necessary for the actions to make sense). We also hope to investigate approaches that better capture narrative necessity without being prohibitively expensive to calculate.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant 1911053 and 2145153. Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Ammanabrolu, P.; Cheung, W.; Broniec, W.; and Riedl, M. O. 2021. Automated storytelling via causal, commonsense plot ordering. In *Proceedings of the 35th AAAI conference on Artificial Intelligence*, 5859–5867.

Bonet, B.; and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence*, 129(1): 5–33.

Cassell, B.; and Young, R. 2013. Ember, toward saliencebased cinematic generation. In *Proceedings of the 9th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 78–81.

Cavazza, M.; Charles, F.; and Mead, S. J. 2002. Characterbased interactive storytelling. *IEEE Intelligent Systems special issue on AI in Interactive Entertainment*, 17(4): 17–24.

Evans, R.; and Short, E. 2013. Versu–a simulationist storytelling system. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(2): 113–130.

Farrell, R.; and Ware, S. G. 2020. Narrative planning for belief and intention recognition. In *Proceedings of the 16th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 52–58.

Farrell, R.; Ware, S. G.; and Baker, L. J. 2020. Manipulating narrative salience in interactive stories using Indexter's Pairwise Event Salience Hypothesis. *IEEE Transactions on Games*, 12(1): 74–85.

Gerrig, R. J. 1993. *Experiencing narrative worlds: on the psychological activities of reading*. Yale University Press.

Helmert, M. 2006a. The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26: 191–246.

Helmert, M. 2006b. New complexity results for classical planning benchmarks. In *Proceedings of the 16th International Conference on Automated Planning and Scheduling*, 52–62.

Hoffmann, J.; and Nebel, B. 2001. The FF planning system: fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14: 253–302.

Kartal, B.; Koenig, J.; and Guy, S. J. 2014. User-Driven Narrative Variation in Large Story Domains using Monte Carlo Tree Search. In *Proceedings of the international conference on Autonomous Agents and Multiagent Systems*, 69–76.

Kreminski, M.; and Mateas, M. 2021. A coauthorshipcentric history of interactive emergent narrative. In *Proceedings of the 14th International Conference on Interactive Digital Storytelling*, 222–235.

Mateas, M.; and Stern, A. 2002. A behavior language for story-based believable agents. *IEEE Intelligent Systems*, 17(4): 39–47.

McCoy, J.; Treanor, M.; Samuel, B.; Reed, A. A.; Mateas, M.; and Wardrip-Fruin, N. 2014. Social story worlds with Comme il Faut. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, 6(2): 97–112.

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.

Porteous, J.; Charles, F.; and Cavazza, M. 2013. NetworkING: using character relationships for interactive narrative generation. In *Proceedings of the international conference on Autonomous Agents and Multi-Agent Systems*, 595–602.

Rabin, S. 2013. *Game AI pro: collected wisdom of game AI professionals*. CRC Press.

Riedl, M. O. 2004. *Narrative planning: balancing plot and character*. Ph.D. thesis, North Carolina State University.

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

Russell, S. J.; and Norvig, P. 2009. *Artificial Intelligence: a modern approach*. Pearson, 3 edition.

Sanghrajka, R.; Young, R. M.; and Thorne, B. 2022. Headspace: incorporating action failure and character beliefs into narrative planning. In *Proceedings of the 18th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 171–178.

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.

Siler, C.; and Ware, S. G. 2022. Solution density and search strategy in narrative generation. *IEEE Transactions on Games*, 14(4): 715–724.

Tapiero, I.; van den Broek, P.; and Quintana, M.-P. 2002. The mental representation of narrative texts as networks: the role of necessity and sufficiency in the detection of different types of causal relations. *Discourse Processes*, 34(3): 237–258.

Teutenberg, J.; and Porteous, J. 2015. Incorporating global and local knowledge in intentional narrative planning. In *Proceedings of the 2015 international conference on Autonomous Agents and Multiagent Systems*, 1539–1546.

Tomaszewski, Z. 2011. On the use of reincorporation in interactive drama. In *Proceedings of the 4th workshop on Intelligent Narrative Technologies at the 7th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 84–91.

Trabasso, T.; and Sperry, L. L. 1985. Causal relatedness and importance of story events. *Journal of Memory and language*, 24(5): 595–611.

Ware, S. G. 2014. *A plan-based model of conflict for narrative reasoning and generation*. Ph.D. thesis, North Carolina State University.

Ware, S. G.; Garcia, E. T.; Fisher, M.; Shirvani, A.; and Farrell, R. 2022. Multi-agent narrative experience management as story graph pruning. *IEEE Transactions on Games*. (forthcoming).

Ware, S. G.; and Siler, C. 2021. Sabre: a narrative planner supporting intention and deep theory of mind. In *Proceedings of the 17th AAAI International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 99–106.

Ware, S. G.; and Young, R. M. 2014. Glaive: a state-space narrative planner supporting intentionality and conflict. In *Proceedings of the 10th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 80–86.

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.; Harrison, B.; and Roberts, D. L. 2014. A computational model of narrative conflict at the fabula level. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, 6(3): 271–288.

Ye, A.; Cui, C.; Shi, T.; and Riedl, M. O. 2022. Neural Story Planning. *arXiv preprint arXiv:2212.08718*.

Young, R. M. 1999. Notes on the use of plan structures in the creation of interactive plot. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, 164–167.

Young, R. M.; Ware, S. G.; Cassell, B. A.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, 37(1-2): 41–64.

Zwaan, R. A.; and Radvansky, G. A. 1998. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2): 162.