# Speeding Up Narrative Planning Using Fog of War Pruning

**Lasantha Senanayake, Stephen G. Ware**

Narrative Intelligence Lab, University of Kentucky
Lexington, Kentucky 40506 USA
lasantha.senanayake@uky.edu, sgware@cs.uky.edu

## Abstract

Narrative planning is the process of generating sequences of actions that form coherent and goal-oriented narratives. Classical implementations of narrative planning rely on heuristic search techniques to offer structured story generation, but often struggle with scalability because of large branching factors and deep search requirements. To improve the speed of narrative planning, we introduce *Fog of War Pruning*, where actions are only allowed if they involve people, places, and things that the protagonist character has discovered. This pruning technique restricts the planning to what is known from the perspective of the story's central character or characters, pruning branches of the search tree that involve actions beyond their current knowledge. This method is particularly useful in narratives where there is a strong protagonist focus and the story unfolds gradually as the character learns. This enables more efficient planning, while more closely aligning with how people would experience stories. Experiments across many narrative domains show that this technique speeds up the search process under identical search limits and lets the planner solve more unique problems.

## Introduction

Narrative planning is a tool used to generate and regenerate structured stories during runtime for interactive narrative environments, such as games and training simulations, that need to maintain a coherent storyline while adapting to user choices (Young et al. 2013). A narrative planner generates stories by reasoning about the logical structure of events (Young 1999). Unlike reactive systems that make limited look-ahead decisions, planning-based approaches evaluate many possible story paths far into the future to ensure both quality and structure. However, the high computational cost of planning (Helmert 2006) means that story generation does not scale well to complex and longer story domains.

Classical heuristic search methods, like those using the $h^+$ and $h^{\max}$ heuristics (Bonet and Geffner 2001), and even search methods specifically designed for narrative planning, like the Glaive heuristic (Ware and Young 2014), can help mitigate some scaling issues. Prior work has also looked at guiding search by using large language models as heuristics (Senanayake and Ware 2025). Still, even with good heuristics, deeply nested searches will eventually become intractable.

Inspired by the *Fog of War* in strategy video games, where unexplored places and characters are hidden until discovered, we introduce Fog of War Pruning for narrative planning. One element of the story, typically the protagonist, begins as seen. When a seen element interacts with other elements, those other elements become seen. Actions involving only unseen things are not allowed. Just like in these games where the fog of war fades as the player explores the map, our approach gradually reveals the elements of the story and only reasons about them once they are discovered.

We implement Fog of War Pruning in the Sabre narrative planner (Ware and Siler 2021) and test it on a suite of benchmark problems. Our pruning technique typically improves the number of problems a planner can solve, the number of nodes visited, and the time spent searching.

Our findings suggest that Fog of War Pruning is especially effective in first person narratives with a central protagonist. It relies on the assumption that characters and the important items in the story are strategically placed for the protagonist to discover, so the search is sometimes incomplete, but when these assumptions hold, the speed and efficiency of the planner is improved.

## Related Work

Early interactive storytelling systems like Tale-Spin (Meehan 1977), Universe (Lebowitz 1985), and Façade (Mateas and Stern 2005) used symbolic logic to represent characters, locations, and conditions. They employed preconditions and effects to control when actions could occur and how they changed the story state, but they did not perform extensive planning searches. Tale-Spin and Universe lacked true planning capabilities, and Façade's planning was limited.

Planning techniques can play an important role in story-driven games as they can provide a clear way to represent causality and event ordering (Young 1999). Off-the-shelf classical planners have been used to make playable interactive versions of well-known narratives such as *Friends* (Cavazza, Charles, and Mead 2002), *Madame Bovary* (Pizzi and Cavazza 2007), and *The Merchant of Venice* (Porteous, Cavazza, and Charles 2010).

Narrative planners extend classical planning by reasoning about story properties. Systems like IPOCL (Riedl and Young 2010) model character intentions, while IMPRACTical (Teutenberg and Porteous 2013) and Glaive (Ware and Young 2014) model conflicts and failed plans. Planners like HeadSpace (Sanghrajka, Young, and Thorne 2022), Ostari (Eger and Martens 2017) and Sabre (Ware and Siler 2021) model characters with mistaken beliefs which enables more complex and realistic story behaviors. A survey by Young et al. (2013) covers various story planning systems. Narrative planning has been used for interactive games (Ware et al. 2014, 2022) and training simulations (Garcia, Ware, and Baker 2019; Fisher, Siler, and Ware 2022).

Because the number of possible action sequences that need to be considered explodes quickly, even for small problems, much research has been done on fast planning algorithms. Much of it has focused on heuristic search, either using traditional planning heuristics (Bonet and Geffner 2001) or methods tailored to stories (Teutenberg and Porteous 2013; Ware and Young 2014).

It is also possible to improve the speed of narrative planning by examining the structure of a story. Recent work used causal necessity as a step-cost function (Ware, Senanayake, and Farrell 2023; Birchmeier and Ware 2025). An action is causally necessary if omitting it would make the plan impossible. Favoring plans rich in causally necessary actions can accelerate the search process. In other words, using causal necessity as a cost function helps prune irrelevant branches and focuses the search on actions that are necessary for plot progression, thereby speeding up the search. The Fog of War Pruning technique we introduce in this paper can be seen as a kind of structural pruning.

Most relevant to this paper are search methods based on *salience*, or how easy it is to remember certain elements of a story (Zwaan and Radvansky 1998). Farrell, Ware, and Baker (2020) did a series of studies that measured how salient each event is to a story based on its protagonist, time, place, causal structure, and goal. Later work built on this idea by using salience as a cost function—plans that link high salience events are rewarded and branches that cannot reach a minimum salience are pruned (Ware and Farrell 2022). Fog of War Pruning can be considered a type of salience-based pruning, though we use a simpler, binary model of salience. One element of the story begins as salient, and each time a salient thing appears in an action it makes the other elements of the action permanently salient also.

Another related line of research on narrative focalization (Bae, Cheong, and Young 2011) limits the narration to a particular character's viewpoint. When the protagonist begins as the only salient object, the early events of a story told with Fog of War Pruning are focalized on the protagonist. However, as the fog clears, events not involving the protagonist quickly become eligible to add to the story.

Another connection can be made to planning under partial observability. Bonet et al. (2011) showed that classical re-planning techniques can be adapted to domains where the agent has only limited information about the world. Our Fog of War Pruning method restricts planning to the protagonist's vantage point, effectively modeling a form of partial observability.

## Example Story Planning Problem

We will use the *Save Gramma* interactive storytelling problem (Ware et al. 2022) throughout this paper to illustrate our method. The player character Tom begins at his sick grandmother's cottage holding a coin. A bandit waits in the nearby camp with a sword and is willing to commit crimes to get rich. A merchant stands at the market with a healing potion, eager for profit but unwilling to break the law. The cottage, camp, and market are linked by a central crossroads. Tom's goal is to obtain the potion and heal his grandmother or die trying.

## Background

In forward state-space planning (Russell and Norvig 2009), the planner begins in the initial world state and adds one fully ground action after another to the end of the plan until a goal condition is reached. In a classical planning problem, any action sequence that reaches a goal state is a valid plan.

Narrative planning typically imposes additional criteria on which plans are considered solutions. We tested our Fog of War Pruning method in the Sabre narrative planner (Ware and Siler 2021), which defines a narrative planning problem as having these elements:

- **Objects** are entities that exist in the story world including characters, places and items. Tom, the cottage, and the healing potion are examples of objects.
- **Fluents** are variables that can be assigned Boolean, numeric, or nominal values. Propositions can be formed by making statements about the value of a fluent or what a character believes the value of a fluent to be. Tom's location is an example of a fluent, and its possible values are the cottage, market, camp, and crossroads. "Tom believes the merchant is at the market," is an example of a belief proposition.
- **Actions** are events similar to ADL operators (Pednault 1994). An action has a precondition that must be true before it can occur, an effect which changes that state, a list of consenting characters who are responsible taking the action, and a function that determines which characters observe an action occurring. Every action has a unique *signature*, which is an action name followed by a list of objects. Tom walking from his cottage to the crossroads is an example of an action.
- **Triggers** are events that fire automatically as soon as their preconditions are met. They do not have consenting characters or observation functions. Tom observing that the bandit is also at the crossroads is an example trigger.
- **Utility Functions** define numeric preferences over states. The author's utility function encodes the author's desired outcome. For example, the author's utility is usually 0, but goes up to 1 when Tom is dead and up to 2 when Tom has the potion at his cottage. Each character also has a personal utility function that guides its behavior. For example, the bandit's utility is her number of coins.
- The **Initial State** defines starting values for all fluents, including any wrong beliefs the characters begin with.

Fog of War Pruning could be used by many different narrative planners, and it does not rely on Sabre's particular problem structure or its definition of what is considered a solution, so we will not describe these in detail. Briefly, a solution is any sequence of actions that carries the story from the initial state to one that increases the author's utility, while ensuring that every step is *explained* for all characters who consent to those actions. An action is *explained for a character* when that character believes they can take the action as part of a plan to improve their own personal utility.

In most planners, including Sabre, each action has a unique signature, which is typically a name followed by a list of arguments. Arguments are objects defined in the problem. For example, a typical action signature looks like:

```
walk(Tom, Cottage, Crossroads)
```

The name of this action signature is `walk`, indicating the type of event that is occurring. This signature has three arguments: `Tom`, `Cottage`, and `Crossroads`, which provide additional detail about who is walking where. In this case, the action means that the character Tom is walking from his cottage to the crossroads.

## Fog of War Pruning

Fog of War Pruning focuses narrative planning by defining some plans which the search does not need to consider—i.e. plans that can be pruned from the search. Every story begins with a set of seen objects, typically just the story's protagonist. An action may only occur if at least one of the arguments in its signature has been seen. After an action occurs, all of its arguments are considered seen.

Let plan $\pi$ be a sequence of $n$ actions $\{a_1, a_2, \ldots, a_n\}$. Let $sig(a_i)$ be a function that returns the set of objects in the signature of action $a_i$. Finally, assume that each problem defines $k$ objects $\{\text{protagonist}_1, ..., \text{protagonist}_k\}$ that will begin as seen in the initial state. In our experiments, this initial set of seen objects was typically a single protagonist character, though it could be any arbitrary set of objects.

**Algorithm 1** Fog of War Pruning

**Input:** A plan $\pi = \{a_1, a_2, ..., a_n\}$
**Output:** $\top$ if $\pi$ should be pruned, else $\perp$

$S \leftarrow \{\text{protagonist}_1, \ldots, \text{protagonist}_k\}$ ▷ Initial seen set
**for** each action $a_i \in \pi$
    **if** $S \cap sig(a_i) = \emptyset$ ▷ Prune if all arguments unseen.
        **return** $\top$
    $S \leftarrow S \cup sig(a_i)$ ▷ Add to the seen set.
**return** $\perp$ ▷ Plan is valid under Fog of War.

Algorithm 1 describes our simple pruning procedure. The algorithm will compute the seen set $S$ of all objects that the protagonist characters have encountered in some way, starting with the initial set of seen objects $\{\text{protagonist}_1, \ldots, \text{protagonist}_k\}$. It considers each action $a_i$ in plan $\pi$ one at a time. If it finds an action such that none of the objects in its signature are seen, it returns $\top$ to indicate that the plan should be pruned—i.e. that the planner should stop building on that plan when trying to find a solution. If an action does not cause the plan to be pruned, all objects from its signature are added to $S$. If no actions cause the plan to be pruned, the planner continues the search.

### Examples

We will illustrate Fog of War pruning using two examples from the *Save Gramma* problem mentioned earlier. The first example shows a plan that is not pruned.

**Example 1: Not Pruned**

$S_0 = \{Tom\}$
$a_1$ : `walk(Tom, Cottage, Crossroads)` ✓
$S_1 = \{Tom, Cottage, Crossroads\}$
$a_2$ : `walk(Bandit, Camp, Crossroads)` ✓
$S_2 = \{Tom, Cottage, Crossroads, Bandit, Camp\}$
$a_3$ : `attack(Bandit, Tom, Crossroads)` ✓
$S_3 = \{Tom, Cottage, Crossroads, Bandit, Camp\}$

At first, only the protagonist Tom is in the seen set. The first step is allowed because it includes Tom in its signature. After the first step, the seen set grows to include the cottage and crossroads locations. The second action is allowed because, even though the bandit and camp have not been seen yet, the crossroads has. Taking the second step expands the seen set to include the bandit and camp. The third action is allowed because all three objects in its signature are seen. Since every action in the sequence references at least one object that was previously seen, the plan is not pruned.

Now we will consider a plan that would be pruned.

**Example 2 — branch pruned**

$S_0 = \{Tom\}$
$a_1$ : `walk(Tom, Cottage, Crossroads)` ✓
$S_1 = \{Tom, Cottage, Crossroads\}$
$a_2$ : `give(Merchant, Guard, MSword)` ×

This second example begins with the same first step. The second step, where the merchant gives their sword to the market guard, fails the visibility test because all three of the objects it references are still unseen. Since no object in the action's signature overlaps with the current seen set, the plan is immediately pruned and no further successors of this plan will be generated.

### Incompleteness

While Fog of War pruning has the potential to speed up narrative planning by pruning many plans during search, it may also make the search incomplete. In other words, some previously solvable problems may become unsolvable if pruning removes all plans that can lead to solutions. The effectiveness of pruning depends on which objects are chosen for the initial seen set, how the signatures of actions are authored, and how the objects in a problem are laid out.

For example, it is possible to write the `walk` action in the *Save Gramma* problem so that its signature includes only the destination location:

```
walk(Tom, Crossroads)
```

The precondition can be written to express that Tom must be at any location adjacent to the crossroads without specifying which one in the signature. This would change which actions become seen and how aggressively plans will be pruned. In our evaluation, which we will describe next, Fog of War Pruning rarely made problems unsolvable.

## Evaluation

We evaluated Fog of War Pruning in the Sabre narrative planner, version 0.8, and tested it on a suite of benchmark problems by several authors that have been collected for that planner (Ware and Farrell 2023). All of these problems were created before we developed Fog of War Pruning, so we believe they represent "naturally occurring" problems that have not been unconsciously designed to work well with this pruning technique. The detailed report cited above provides background and historical context for each problem, including their origins in prior narrative planning research. This report serves as a useful reference for understanding the design and motivation behind these problems.

### Benchmark Problems

Table 1: The number of characters, fluents, actions, and triggers in the benchmark narrative planning problems.

| Problem | Problem size | | | |
|---|---|---|---|---|
| | $\lvert C\rvert$ | $\lvert F\rvert$ | $\lvert A\rvert$ | $\lvert T\rvert$ |
| Aladdin | 5 | 150 | 282 | 378 |
| Basketball | 4 | 93 | 168 | 192 |
| Bribery | 3 | 16 | 27 | 0 |
| Deer Hunter | 3 | 35 | 28 | 76 |
| Fantasy | 4 | 68 | 76 | 136 |
| Gramma | 4 | 61 | 812 | 896 |
| Hospital | 4 | 57 | 102 | 196 |
| Jailbreak | 3 | 26 | 106 | 54 |
| Lovers | 3 | 40 | 312 | 375 |
| Raiders | 3 | 21 | 39 | 66 |
| Secret Agent | 2 | 14 | 44 | 75 |
| Space | 2 | 23 | 29 | 62 |
| Western | 4 | 99 | 352 | 637 |

The test suite provides 14 benchmark story domains. One of them, *Treasure Island*, contains only 5 ground actions, 3 of which have no arguments in their signatures. It can be solved quickly by any planner configuration, so we exclude it from our evaluation. Table 1 shows the size of each benchmark domain, represented by the number of characters $|C|$, fluents $|F|$, actions $|A|$, and triggers $|T|$ after the problem has been fully grounded and simplified.

For the 12 remaining domains, several define multiple problems that can be solved by specifying higher author utility thresholds for the search. For example, *Gramma Any* sets the author utility threshold for a solution to 1, which will accept any story where Tom dies or succeeds on his mission. *Gramma Win* sets the threshold to 2, which only accepts stories where Tom succeeds. In total, we evaluated a total of 26 narrative planning problems.

### Choosing Protagonists

For each domain, we need to choose one or a few characters to start in the initial seen set. We made these choices based on the descriptions of the benchmark problems.

- Several problems have a clear protagonist: *Deer Hunter* (Bubba), *Secret Agent* (the Agent), *Aladdin* (Aladdin), *Hospital* (Hathaway), *Western* (Hank), *Space* (Zoe), *Save Gramma* (Tom), *Raiders of the Lost Ark* (Jones).
- For *Bribery*, we used the Villain as the protagonist, since that character is featured most prominently in both example solutions, and the other character, Hero, does not appear in the first example solution.
- For *Basketball* we used the detective, Sherlock, as the protagonist, despite the fact that he is not essential to some solutions.
- For *Fantasy* we used Talia, since the author's utility revolves around her utility.
- For *Jailbreak* we used both Earnest and Roy. This domain was originally designed for an interactive story where either of those characters could become the protagonist depending on choices made by the player, so we decided to use both.
- For *Lovers* we use C1 as the protagonist, since they are the one who needs to lie in the example solution.

### Planner Configurations

We tested many combinations of Sabre search techniques on each benchmark problem. Each configuration was tested with and without Fog of War Pruning.

Breadth-first search (BFS) is a simple search method that always expands a shortest plan first. Here, a "shortest plan" means the one where the fewest actions have occurred in the author's plan since the initial state—what Sabre calls *temporal cost*. Explanation-first search (EFS) requires an action to be explained for all other consenting characters before it is added to a plan. Goal-first search (GFS) requires the planner to verify that a plan can improve utility before it attempts to explain any of its actions. In other words, EFS explores only plans with explained actions while searching for the goal, while GFS explores only plans known to achieve the goal but which may not be composed of explained actions. When faced with multiple plans that are legal to expand next, basic EFS and GFS revert to breadth-first search behavior, expanding a shortest plan first.

We also tested variants of these three methods as heuristic search. BFS, when done with a heuristic, becomes A* search, which always expands a plan that minimizes the sum of its cost and heuristic value, where cost is temporal cost and the heuristic is one of three classical planning heuristics that estimates the distance from the current state to the nearest state where utility is improved. EFS and GFS can also be done using the sum of cost and heuristic to choose which legal plan to expand next.

For all three types of heuristic search (A*, EFS, and GFS), we tested three well-known classical planning heuristics: $h^+$, $h^{\max}$ (Bonet and Geffner 2001), and a relaxed-plan heuristic $h^{\text{rp}}$ which is similar to the Fast-Forward heuristic (Hoffmann and Nebel 2001). The details of these heuristics are not important for this evaluation, but all of them work by relaxing the planning problem, solving the relaxed problem, and using the solution to that relaxed problem as an approximation of the solution to the real problem.

We also placed several limits on each search to improve their chances of success. The Author Temporal Limit (ATL) is the maximum number of actions that may appear in a solution to the problem. The Character Temporal Limit (CTL) restricts how long an individual character's explanatory plan may be. The Epistemic Limit (EL) fixes how many levels of nested beliefs the planner reasons about. We chose these values based on the recommended settings described in the benchmark suite (Ware and Farrell 2023). We also limited each search to visit a maximum of 1 000 000 nodes before failing automatically. Visiting a node means that, for action whose preconditions are satisfied, we add a successor plan to the search queue with that action added to the end (unless doing so would violate one of the above mentioned limits).

When adding actions to a plan, the order in which actions are considered can influence the success rate of a planner configuration. For example, considering `walk` actions before `attack` actions may find solutions faster. For this reason, we ran each planner configuration on each problem 10 times, shuffling the problem's list of actions between each run to control for this tendency.

In total, we tested 3 search techniques (BFS, EFS, GFS) with and without three heuristics ($h^+$, $h^{max}$, $h^{rp}$) for a total of 12 planner configurations. Each was run 10 times on each of 26 problems. Each configuration was tested with and without Fog of War Pruning, for a total of 6 240 tests.

### Hardware

We performed all tests on a computer with an Intel Xeon w3-2425 processor and 512GB RAM. No non-system processes were active during the tests.

## Results

We summarize our results in two tables. Table 2 summarizes the results for each planner across all problems. Table 3 summarizes the results for each problem across all planners.

## Results by Planner

Table 2 shows how Fog of War Pruning affected each planner configuration based on the following features.

- **Problems Solved:** The total number of tests where the planner found a solution. Recall that each planner was tested 10 times on 24 problem, making 240 the maximum value, though no planner achieved this maximum.
- **Unique Problems Solved:** The total number of problems that the planner solved at least once. The maximum value is 24, though no planner achieved this.
- **Average Nodes Visited:** The mean number of plans visited during all searches across all problems. When calculating this average, we only consider tests where the planner succeeded with and without Fog of War Pruning. This allows us to make meaningful comparisons between planners with and without pruning. If we did *not* calculate the average this way, planners that solved more problems might appear to visit more nodes on average, since more tests would have been considered in the calculations.
- **Average Time:** The mean number of milliseconds that a planner spent searching during all tests. Again, we only consider tests where the planner succeeded with and without pruning.

Our results show that Fog of War Pruning improves planner performance for most configurations. Out of the 12 search types, *BFS* and all *GFS* variants profit the most. Specifically in *GFS*, *GFS* $h^{max}$ and *GFS* $h^+$ the number of nodes visited falls by 65-83% and runtime drops by a similar margin while each variant solves between 1 and 5 more unique problems than the baseline. Among A* planners, *A** $h^{max}$ shows the largest gain, solving 2 more unique problems than the baseline and doing so with just 60K nodes visited compared to the baseline's 245K. *A** $h^{rp}$ and *EFS* $h^+$ display the same pattern of faster search with higher coverage.

Fog of War Pruning is less beneficial to *EFS* search variants. Basic *EFS*, *EFS* $h^{max}$ and *EFS* $h^{rp}$ all solve fewer unique problems compared to the baseline, indicating that pruning may be occasionally cutting off valid explanations that these planners rely on. *A** $h^+$ is an unique outlier, Fog of War Pruning does solve more problems than the baseline, though it visits 2 times more nodes and takes 3x more time.

Across all these 12 planners, Fog of War Pruning reduces total node visits by 37% and also reduces the runtime by 48%. It was also able to solve more unique problems. We believe this demonstrates that our pruning technique is broadly advantageous.

## Results by Problem

Table 3 shows how Fog of War Pruning affected all planners that ran on each problem.

- **Time Solved:** The total number of times a test on this problem found a solution. Each problem was tested 10 times by 12 planners, so 120 is the maximum value.
- **Unique Planners Solved:** The number of unique planner configurations that were able to solve this problem at least once. There are 12 planners, so the maximum value is 12.
- **Average Nodes Visited:** The mean number of plans visited by planners solving this problem. When calculating this average, we only consider tests where the planner succeeded with and without Fog of War Pruning. For example, there is no value for the *Fantasy All* problem because, while some planners were able to solve it without pruning and some were able to solve it with pruning, there was no overlap; i.e. there were no planners

Table 2: Planner performance across all problems with and without Fog of War Pruning. Green cells indicate values that improved with pruning; red cells indicate values that got worse; yellow cells indicate no change.

| Planner | Without Fog of War Pruning | | | | With Fog of War Pruning | | | |
|---|---|---|---|---|---|---|---|---|
| | Solved | Unique | Nodes | Time (ms) | Solved | Unique | Nodes | Time (ms) |
| BFS | 180 | 18 | 991 674 | 136 882 | 182 | 19 | 248 208 | 31 303 |
| A* $h^+$ | 170 | 17 | 425 938 | 328 356 | 182 | 19 | 1 014 422 | 1 279 811 |
| A* $h^{max}$ | 190 | 19 | 244 834 | 239 237 | 210 | 21 | 60 076 | 63 143 |
| A* $h^{rp}$ | 200 | 20 | 1 570 002 | 3 751 521 | 206 | 22 | 1 477 873 | 1 814 124 |
| EFS | 161 | 17 | 2 690 414 | 1 489 233 | 150 | 15 | 911 580 | 74 432 |
| EFS $h^+$ | 163 | 17 | 1 408 877 | 776 121 | 170 | 17 | 569 448 | 213 419 |
| EFS $h^{max}$ | 210 | 21 | 366 407 | 466 880 | 180 | 18 | 365 632 | 364 860 |
| EFS $h^{rp}$ | 177 | 19 | 1 948 922 | 919 390 | 171 | 18 | 1 649 025 | 705 464 |
| GFS | 140 | 14 | 439 761 | 53 405 | 152 | 16 | 376 162 | 25 668 |
| GFS $h^+$ | 148 | 15 | 635 002 | 50 175 | 160 | 16 | 222 838 | 17 403 |
| GFS $h^{max}$ | 160 | 16 | 517 134 | 983 199 | 210 | 21 | 87 958 | 156 644 |
| GFS $h^{rp}$ | 160 | 16 | 559 227 | 221 751 | 179 | 18 | 495 899 | 133 414 |
| **Total** | 2 059 | 209 | 11 798 190 | 9 416 150 | 2 152 | 220 | 7 479 120 | 4 879 684 |

Table 3: Results for each problem across all planners with and without Fog of War Pruning. Only instances where at least one configuration produced a result are listed. Green cells indicate values that improved with pruning; red cells indicate values that got worse; yellow cells indicate no change.

| Problem | Without Fog of War Pruning | | | | With Fog of War Pruning | | | |
|---|---|---|---|---|---|---|---|---|
| | Solved | Unique | Nodes | Time (ms) | Solved | Unique | Nodes | Time (ms) |
| Basketball Any | 109 | 12 | 1 052 472 | 497 927 | 80 | 10 | 3 583 962 | 2 636 386 |
| Basketball Both | 20 | 2 | 487 291 | 905 462 | 3 | 2 | 991 893 | 1 252 493 |
| Bribery | 120 | 12 | 34 566 | 718 | 120 | 12 | 22 887 | 339 |
| Deer Hunter Any | 120 | 12 | 197 583 | 22 488 | 120 | 12 | 30 831 | 3 273 |
| Deer Hunter Both | 50 | 5 | 1 547 386 | 3 698 207 | 100 | 10 | 539 875 | 61 796 |
| Fantasy Any | 120 | 12 | 752 | 245 | 120 | 12 | 393 | 151 |
| Fantasy Two | 120 | 12 | 141 607 | 54 244 | 120 | 12 | 38 929 | 16 649 |
| Fantasy All | 10 | 1 | – | – | 40 | 4 | – | – |
| Gramma Any | 120 | 12 | 170 870 | 109 823 | 120 | 12 | 92 883 | 71 971 |
| Gramma Win | 100 | 10 | 1 708 211 | 1 970 295 | 110 | 11 | 224 166 | 240 734 |
| Hospital Any | 98 | 10 | 1 258 612 | 604 292 | 110 | 11 | 734 476 | 346 220 |
| Jailbreak Escape | 50 | 5 | 1 576 130 | 475 836 | 70 | 7 | 343 880 | 79 529 |
| Jailbreak Lose | 110 | 11 | 1 346 465 | 60 710 | 120 | 12 | 498 541 | 25 071 |
| Jailbreak Revenge | 10 | 1 | – | – | 37 | 4 | – | – |
| Lovers | 61 | 7 | 2 242 408 | 1 010 691 | 82 | 9 | 318 824 | 133 953 |
| Raiders | 120 | 12 | 10 958 | 1 051 | 120 | 12 | 3 440 | 399 |
| Secret Agent | 120 | 12 | 3 054 | 86 | 120 | 12 | 932 | 34 |
| Space Any | 120 | 12 | 88 | 15 | 120 | 12 | 91 | 19 |
| Space Two | 120 | 12 | 181 | 33 | 120 | 12 | 95 | 18 |
| Space Three | 120 | 12 | 365 | 81 | 120 | 12 | 95 | 19 |
| Space Four | 120 | 12 | 7 583 | 1 577 | 120 | 12 | 1 095 | 195 |
| Space All | 120 | 12 | 11 610 | 2 368 | 80 | 8 | 51 832 | 10 437 |
| Western | 1 | 1 | – | – | 0 | 0 | – | – |
| **Total** | 2 059 | 209 | 11 798 190 | 9 416 150 | 2 152 | 220 | 7 479 120 | 4 879 684 |

that solved it with and without pruning, so we cannot make a meaningful comparison here.

- **Average Time:** The mean number of milliseconds that planners spent working on this problem. Again, we only consider tests where the planner succeeded with and without pruning.

Across the benchmark suite, Fog of War Pruning improves both nodes visited and runtime for the majority of problems. In large narrative spaces like *Deer Hunter Both*, *Fantasy All*, and *Jailbreak Revenge* 3 to 5 additional planners now find solutions. In some problems, these gains come with dramatic speed-ups. *Deer Hunter Both* cuts the node expansion by 65% and runtime is reduced by 98% in comparable runs. In problems like *Lovers*, *Hospital Any*, and *Gramma Win*, Fog of War Pruning raises the success rate by 10-20% while decreasing visited nodes and runtime.

Several problems that were already solved in the baseline (*e.g.*, *Fantasy* and in several *Space* problems) see no change in coverage but yield noticeable efficiency gains.

Some problems display the opposite behavior. The two *Basketball* problems and *Space All* lose 25–40 successful runs with Fog of War Pruning. The single successful baseline search method in *Western* (which was solved only 1 time out of 10 by only one planner) disappears with pruning. These problems suggest that certain problem structures rely on "off screen" events the protagonist has not yet witnessed and pruning those branches may make it more difficult or impossible to find solutions.

These results by problem further confirm that Fog of War Pruning can benefit a wide variety of narrative planning problems.

### Success Rate

Table 4 breaks down how often Fog of War Pruning affects the success rate of a planner on a problem. We consider 12 planners on 26 problems, which provides a total of $12 \times 26 = 312$ opportunities to observe a change in success rate. Of those, pruning changes the success rate 39 times (12.5%). In the other 273 cases (87.5%), the success rate of that planner on that problem stayed the same.

For the 39 affected cases:

- In 24 cases, Fog of War Pruning improves the success rate, and in 18 cases problems that were unsolvable without pruning are now 100% solvable with pruning.
- In 15 cases, success rate decreased, and in 7 cases, fully solvable problems became unsolvable.

In other words (as summarized in Table 5), Fog of War Pruning improves success rate in 7.7% of the total cases, and reduces success rate 4.8% of the total cases. In most cases, success rate is not affected.

### Plan Length

Sometimes Fog of War Pruning affects the length of solutions that a planner finds. All things being equal, we prefer shorter plans over longer plans.

Recall that we shuffle the order of actions in the problem between each of the 10 times that we tested the same planner on the same problem. In some cases, the same planner might find a solution of a different length to the same problem. For example, the first time a planner is tested on a problem it might find a solution with 5 actions. Then, after actions have been shuffled, the same planner might find a different solution with 6 actions to the same problem. This means that we can measure the minimum plan length, maximum plan length, and average plan length for each planner/problem pair, and there are 312 such cases.

Table 6 summarizes how Fog of War Pruning affected the min, max, and average plan length in cases that could be solved both with and without pruning. In 3 cases, the minimum plan length decreased (which is good). In 20 cases, the minimum plan length increased (which is bad). In 5 cases, the maximum plan length decreased. In 20 cases, the maximum plan length increased. In 5 cases the average plan length decreased. In 24 cases, the average plan length increased. Fog of War Pruning rarely affects the length of solutions found, but when it does, it is more likely to increase the plan length.

### Runtime

Table 7 summarizes how often Fog of War Pruning affected runtime. Again, we consider 12 planners on 26 test problems for 312 opportunities to observe a change in runtime. To make a controlled comparison, we will only consider cases where the success rate was the same with and without pruning, which was 212 cases.

In the 212 cases where Fog of War Pruning did not affect the success rate of a planner, it improved runtime 75.5% of the time. This indicates that our pruning method usually speeds up a planner.

### Overall Performance

Table 8 summarizes the performance of Fog of War Pruning in all 312 cases. Pruning raises performance (by either improving success rate or visiting fewer nodes) far more often than it lowers performance (by either reducing success rate or visiting more nodes). It raises performance in 172 cases, or 55.1% of the total. It only lowers performance in 43 cases, or 13.8% of the total.

Since there are 82 cases where the planner found no solutions with or without pruning, we can remove those cases from consideration and reconsider these results. From this perspective, pruning raises performance 74.5% of the time and only lowers it 18.7% of the time.

## Limitations

As mentioned previously, perhaps the most significant limitation of Fog of War Pruning is that it can cause a solvable problem to become unsolvable. This happens when, for example, a story requires an important but unobserved event to happen at the start of the story.

Consider the *Space* domain, where protagonist Zoe starts in her starship and wants to explore a nearby but geologically unstable planet. The shortest solution to this problem (that achieves author utility 1) is:

```
begin_erupt(Surface)
erupt(Surface)
```

Table 4: Changes in solvability when applying Fog of War Pruning, compared with a no pruning baseline, across 312 tests.

| Outcome | # comparisons | % of total (312) |
|---|---|---|
| Unsolvable → fully solvable (now 100 %) | 18 | 5.8 |
| Partly solvable → fully solvable | 3 | 1.0 |
| Unsolvable → *partly* solvable ($<100\%$) | 3 | 1.0 |
| Still partly solvable, success rate increased | 0 | 0.0 |
| Still partly solvable, success rate decreased | 1 | 0.3 |
| Fully solvable → partly solvable | 4 | 1.3 |
| Fully solvable → unsolvable | 7 | 2.2 |
| Partly solvable → unsolvable | 3 | 1.0 |
| No change in solvability | 273 | 87.5 |

† **Fully solvable**: planner succeeds on all 10 test runs. **Partly solvable**: planner succeeds on at least one but not all runs (1–9 / 10). **Unsolvable**: planner fails on every run (0 / 10).

Table 5: Effect of Fog of War Pruning on success rate, compared with a no pruning baseline, across 312 tests.

| Outcome | # of comparisons | % of total (312) |
|---|---|---|
| Success rate increased[1] | 24 | 7.7% |
| Success rate decreased[2] | 15 | 4.8% |
| Success rate unchanged | 273 | 87.5% |

[1] Example: Baseline solves 2 times out of 10 runs, but with pruning, planner solves 7 out of 10 runs.
[2] Example: Baseline solves 9 times out of 10 runs, but with pruning, planner only solves 6 out of 10 runs.

Table 6: Change in plan length when applying Fog of War Pruning, evaluated only on tests where *both* pruning and non-pruning planners produced at least one solution.

| Outcome category | Min | Max | Avg |
|---|---|---|---|
| # of comparisons where the plan length decreased with pruning | 3 | 5 | 5 |
| # of comparisons where the plan length increased with pruning | 20 | 20 | 24 |

†**Min**: shortest plan length among the 10 runs. **Max**: longest plan length among the 10 runs. **Avg**: mean over all successful runs (failed runs excluded).

It seems like this solution cannot be found by Fog of War Pruning because it does not involve Zoe. However, we were pleasantly surprised to discover that this limitation is not as problematic as we first suspected.

While the above 2 action solution would indeed get pruned, the following 3 action solution can still be found:

```
teleport_from_ship(Zoe, Ship, Surface)
begin_erupt(Surface)
erupt(Surface)
```

This plan achieves an author utility of 2, so it can be used as a solution to both *Space Any* and *Space Two*. In addition, Sabre has a feature which requires that every action in a solution be non-redundant (meaning no actions can be left out). To check this requirement, Sabre will expand every strict subsequence of the solution. This means that Sabre will expand the original 2 action solution in the process of checking the 3 action solution, so Sabre is actually still able to find both solutions, even with Fog of War Pruning (though it will need to expand more nodes when pruning).

This means that Fog of War Pruning only prevents Sabre from finding a solution when the planner would prune not only that solution but also every plan that contains the solution as strict subsequence. In practice, this seems to happen rarely. When it does happen, those solutions can again be found by simply expanding the initial seen set for a domain, though this will reduce the effectiveness of pruning.

Fog of War Pruning can also hinder proactive mediation. Because the pruning selects actions that intersect the player's current seen set, setup steps that would prevent future problems (e.g., staging an item, relocating an agent, or triggering a environment change) are usually pruned.

## Conclusion

Fog of War Pruning is a simple technique for ruling out stories that include actions composed only of unseen elements. We believe our results demonstrate that it typically improves narrative planner performance across most planner configurations and most problems.

Across all planners, the total number of nodes visited dropped from about 11.8 million in the baseline with no

Table 7: Runtime impact of Fog of War Pruning versus a no-pruning baseline on 312 tests. "Adjusted %" is computed over the 212 cases where runtime was measured in both configurations.

| Outcome | # of tests | % of total (312) | Adjusted % (of 212) |
|---|---|---|---|
| Pruning finished faster than the baseline. | 160 | 51.3% | 75.5% |
| Pruning finished slower than the baseline. | 52 | 16.7% | 24.5% |

Runtime was not measured for 100 comparisons (32.1%) because the success rate differed between configurations or both searches failed.

Table 8: Summary of the overall effect of Fog of War Pruning versus a no-pruning baseline on 312 tests. "Adjusted %" is computed over the 230 cases where at least one configuration produced a solution.

| Outcome | # of tests | % of total (312) | Adjusted % (of 230) |
|---|---|---|---|
| Higher success rate *or* fewer nodes visited with pruning. | 172 | 55.1% | 74.8% |
| Lower success rate *or* more nodes visited with pruning. | 43 | 13.8% | 18.7% |
| No change in performance | 15 | 4.8% | 6.5% |
| No solution in either configuration | 82 | 26.3% | — |

pruning to 7.5 million with pruning, and total runtime roughly halved (from $9.42 \times 10^5$ ms to $4.88 \times 10^5$ ms). Pruning allowed planners to solve more problems. Cumulatively, pruning solved 220 of 312 unique tasks (2152 successful runs), versus the baseline's 209 (2059 runs).

In many cases, Fog of War Pruning reduced search effort on solved instances (fewer nodes visited in less time) while often increasing coverage (more "Solved" and "Unique" tasks). For example, Breadth-first search succeeded on 182 tests with pruning (vs. 180 without) and expanded only $\sim$25% as many nodes ($\approx$248k vs. $\approx$992k), running in $\sim$23% of the original time. Goal-first searches all benefited; GFS with the $h^{\max}$ heuristic saw the largest gains (210 tests solved vs. 160, and only $\sim$17% of the original nodes visited), and other GFS variants also solved more problems while cutting runtime roughly in half.

All of the A* planners likewise improved. A* $h^{\max}$ solved 210 problems (vs. 190) with about 75% fewer node expansions and time, and A* with a relaxed plan heuristic solved 206 (vs. 200) with roughly 50% less runtime. In short, Fog of War Pruning allowed most planners to reach goals that were previously unreachable, and to do so faster.

While pruning was generally beneficial, it had some limitations. In a few cases, pruning was too aggressive and reduced completeness. Notably, Explanation-first Search lost coverage; it succeeded on only 150 tests with pruning (vs. 161 without). EFS $h^{\max}$ saw a larger drop (180 vs. 210 tests). This suggests that Fog of War Pruning sometimes prunes paths leading to valid solutions in these planners.

Aside from EFS search cases, no planner became worse and every planner either maintained or increased its number of tests solved, and for the tasks it did solve Fog of War Pruning never increased effort, except for A* with $h^+$.

## Future Work

We believe Fog of War Pruning not only accelerates planning but also does a better job of simulating the mental model of a player in an interactive story game. We think it produces more salient stories by pruning any action that involves a person, place, or object outside the player's known set. The planner naturally produces plans that are better explained from the protagonist's viewpoint.

For instance, consider a scene where Tom (the protagonist) has not yet seen a bandit lurking at a crossroads. Without pruning, the planner might direct the bandit to move to the crossroads before Tom arrives. With pruning, that plan is rejected, and instead the bandit's appearance is delayed until after Tom reaches the crossroads. In the second plan, every action is consistent with what Tom knows at the time it occurs. We believe that this alignment between plan steps and character knowledge is a promising step toward more coherent story generation.

In future work, we aim to test whether Fog of War Pruning improves narrative quality and whether it leads to more believable character behavior in games and simulations.

## Code

To enable replication and to encourage other researchers to build on this work, we have made our code, evaluation scripts, and other materials available on GitHub, allowing others to repeat our experiments or test different planners.

https://github.com/lazzy07/fog_of_war_pruning

## Acknowledgments

This material is based upon work supported by the U.S. National Science Foundation under Grant No. 2145153 and the U.S. Army Research Office under Grant No. W911NF-24-1-0195. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Army Research Office.

## References


Bae, B.-C.; Cheong, Y.-G.; and Young, R. M. 2011. Toward a computational model of focalization in narrative. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, FDG '11, 313–315. New York, NY, USA: Association for Computing Machinery. ISBN 9781450308045.

Birchmeier, G.; and Ware, S. G. 2025. Speeding up narrative planning with Causal Width Search and Pruning. In *Proceedings of the 21st AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*. (forthcoming).

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

Bonet, B.; and Geffner, H. 2011. Planning under partial observability by classical replanning: theory and experiments. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, 1936–1941. AAAI Press. ISBN 9781577355151.

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

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

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.

Fisher, M.; Siler, C.; and Ware, S. G. 2022. Intelligent de-escalation training via emotion-inspired narrative planning. In *Proceedings of the 13th Intelligent Narrative Technologies workshop at the 18th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*.

Garcia, E. T.; Ware, S. G.; and Baker, L. J. 2019. Measuring presence and performance in an intelligent virtual reality police use of force training simulation prototype. In *Proceedings of the 32nd AAAI international conference of the Florida Artificial Intelligence Research Society*, 276–281.

Helmert, M. 2006. 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.

Lebowitz, M. 1985. Story-telling as planning and learning. *Poetics*, 14(6): 483–502.

Mateas, M.; and Stern, A. 2005. Structuring content in the Façade interactive drama architecture. In *Proceedings of the 1st AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 93–98.

Meehan, J. R. 1977. TALE-SPIN, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 91–98.

Pednault, E. P. D. 1994. ADL and the state-transition model of action. *Journal of Logic and Computation*, 4(5): 467–512.

Pizzi, D.; and Cavazza, M. 2007. Affective storytelling based on characters' feelings. In *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*, 111–118.

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. 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. R. 2022. HeadSpace: Incorporating Action Failure and Character Beliefs into Narrative Planning. In *Artificial Intelligence and Interactive Digital Entertainment Conference*.

Senanayake, L.; and Ware, S. G. 2025. Language Models as Narrative Planning Heuristics. In *Proceedings of the 20th International Conference on the Foundations of Digital Games*, 1–9.

Teutenberg, J.; and Porteous, J. 2013. Efficient intent-based narrative generation using multiple planning agents. In *Proceedings of the 2013 international conference on Autonomous Agents and Multiagent Systems*, 603–610.

Ware, S. G.; and Farrell, R. 2022. Salience as a narrative planning step cost function. In *Proceedings of the IEEE Conference on Games*, 433–440.

Ware, S. G.; and Farrell, R. 2023. A Collection of Benchmark Problems for the Sabre Narrative Planner. Technical report, Narrative Intelligence Lab, University of Kentucky.

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*, 15(3): 378–387.

Ware, S. G.; Senanayake, L.; and Farrell, R. 2023. Causal Necessity as a Narrative Planning Step Cost Function. In *Proceedings of the 19th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 155–164.

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.; Young, R. M.; Stith, C.; and Wright, P. 2014. The Best Laid Plans. http://cs.uky.edu/~sgware/projects/blp/.

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.