2022 IEEE Conference on Games (CoG)
Beijing, China, August 21-24, 2022

Salience as a Narrative Planning Step Cost Function

Stephen G. Ware
Department of Computer Science
University of Kentucky
Lexington, KY, USA
sgware@cs.uky.edu

Rachelyn Farrell
Department of Computer Science
University of Kentucky
Lexington, KY, USA
rachelyn.farrell@uky.edu

2022 IEEE Conference on Games (CoG) | 978-1-6654-5989-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/CoG51982.2022.9893667

***Abstract*—Psychological research has demonstrated that as we experience a story several features affect the salience of its events in memory. These features correspond to *who? where? when? how?* and *why?* questions about those events. Computational models of salience have been used in interactive narratives to measure which events people most easily remember from the past and which they expect more readily from the future. We use three example domains to show that events in sequences that are solutions to narrative planning problems are generally more salient with each other, and events in non-solution sequences are less salient with each other. This means that measuring the salience of a sequence of actions during planning can serve as an efficient cost function to improve the speed, and perhaps also the quality, of a narrative planner.**

***Index Terms*—narrative, planning, salience, search**

I. Introduction

Algorithms for generating stories have long been of interest to the entertainment AI community as a tool for controlling interactive experiences, such as video games or training simulations [1]. In previous work, we explored how modeling the salience of events while generating an interactive story can inform the choices offered to the player [2], [3]. In this paper, we explore whether modeling salience can also improve the speed of the search algorithms.

As we experience a narrative, we segment it into discrete events and store information about past events in memory. These past events can be easier or harder to recall based on how they relate to the event or situation we are currently experiencing (dubbed the current *situation model*). The event-indexing model (EIM) [4] has established at least five important indices by which stored events can be linked in memory:

- Protagonist (*who?*): It is easier to recall events that involve characters who are involved in the current event.
- Time (*when?*): It is easier to recall events that happened in the same time frame (e.g. day) as the current event.
- Space (*where?*): It is easier to recall events that occurred in the same location as the current event.
- Causality (*how?*): It is easier to recall events which caused the current event to occur.
- Intentionality (*why?*): It is easier to recall events which happened in service of the same goal as the current event.

This research was supported by the U.S. National Science Foundation, Grant No. IIS-1911053. All opinions are our own.

For example, when the current event of a story is happening to King Arthur (protagonist) in Camelot (space), previous events that also involved him and took place in Camelot are more salient, or easier to recall, than they would be otherwise.

These five indices can be mapped onto features readily available in planning-based computational models of narrative [5]. A planning algorithm is one that searches for a sequence of events to achieve a goal. Planning algorithms are a popular tool for story generation because they offer a formal, generative model of a goal-directed sequence of events, along with the causal and temporal constraints on the events [6]. The knowledge representation for planning algorithms has been extended to include models of character goals [7], [8], character beliefs [9], [10], conflict [11], and others (see Young et al. [1] for a survey). This rich knowledge representation provides a way to model the *who*, *when*, *where*, *how*, and *why* of a plan's events, which can then be used to model salience.

Planning algorithms are especially attractive for interactive narratives where a story must be repaired or regenerated in response to a player's actions. For example, fast planners have been used to make interactive versions of the TV show *Friends* [12], Flaubert's novel *Madame Bovary* [13], Shakespeare's play *The Merchant of Venice* [14], and various role-playing games [15], [16]. Planning is computationally expensive [17], which limits the scope of the stories that can be generated at run time. Algorithms like IMPRACTical [8], [9], Glaive [18], and Sabre [19] adapt research on fast heuristic search to speed up narrative generation.

In this paper we provide preliminary evidence that a simple model of event salience can be used to speed up narrative planning. Algorithms like Breadth-First Search, Uniform Cost Search, and A* all measure some version of a plan's length or cost when deciding which partial plans to extend when searching for a solution. Typically a plan's cost is simply its length (number of events) or the time it takes to execute those events in the case of a scheduler. We use an alternative measure of plan cost based on salience: the more salience indices two events share, the smaller the distance between them, and the cost of a plan is the sum of the distances between contiguous events. Using three domains, we show that solutions tend to have a lower salience cost than non-solutions, and that narrative planners can sometimes find solutions faster by prioritizing plans with low salience cost. We suspect that modeling salience during search can improve both the speed and coherence of interactive stories, though this paper deals only with speed.

II. Related Work

STRIPS-like planning problems [20] define a set of parameterized operators, or templates, to represent the types of events that can occur in the story world. Each operator defines preconditions—logical literals that must be true before the event can happen, and effects—literals that become true as a result of the event. Operators imply a set of ground actions, which are every way specific arguments can be substituted for the operator's parameters. The action $walk(Tom, Cottage, Crossroads, Day_1)$ is a ground action that assigns four specific arguments to the parameters of the operator $walk(person, from, to, date)$.

The problem also defines an initial state, a conjunction of ground literals that completely describes the initial configuration of the world, and a goal, a conjunction of ground literals to be achieved. A solution to the problem is a sequence of actions that can be executed from the initial state to achieve the goal.

The Intentional Partial-Order Causal Link planner (IPOCL) [7] designates some objects in the domain as the set of story characters who can possess their own *character goals*. IPOCL operators define a set of *consenting characters*—those who are responsible for taking the action and must have a reason for doing so. During search, IPOCL tracks *intention frames* for each character—sequences of actions taken by the character to achieve their character goal. To ensure all actions in the solution appear properly motivated, the planner guarantees that each action appears within an intention frame for each of its consenting characters.

Indexter [5] is a computational model that maps the five EIM indices onto IPOCL plans. The characters and character goals introduced by IPOCL are used to determine the protagonist and intentionality indices: Two actions share *protagonist* if the story's protagonist (a predefined character) is a consenting character in both actions, and they share *intentionality* if they appear within the same intention frame. Indexter defines the *causality* index using the planner's causal link structures, which explicitly track how the effects of earlier actions establish preconditions needed by later actions [21]. There is a great deal of nuance to causality and the types of causal relationships between actions that are perceived during narrative processing [22]. For our purposes we use a simple definition of enablement, where two actions are determined to share *causality* if one enables the other.

Indexter extends the IPOCL model to account for the remaining indices by requiring location and time frame parameters for each operator. Thus two actions share *space* if their location parameter is the same symbol, and *time* if their time frame parameter is the same symbol. The Indexter model was shown to be sufficiently accurate to validate the pairwise event salience hypothesis—that past actions are more salient when they share indices with the current action—using readers' response times as an indicator of salience [23]. Readers were interrupted during a story and asked to recall previous actions, and actions which shared indices with the current action were recalled faster (e.g. when the most recently read action involved King Arthur, readers were able to recall past actions involving Arthur faster).

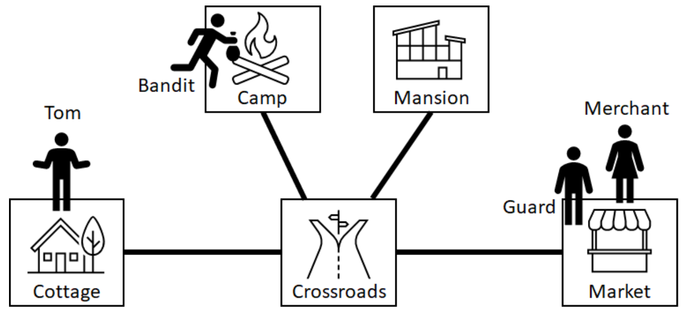

Fig. 1: The *Grammalot* domain's characters and locations. Note the merchant starts at the market but may move to her mansion at night.

Other studies suggest that this model of salience can be used to represent meaningful dimensions by which humans segment perceived trajectories through interactive narrative spaces. In one study, subjects reported feeling more agency when they were making choices whose outcomes differed along at least one index [2]. In another, readers' choices for story endings were predicted based on the number of indices each ending shared with previous choice outcomes [3]. Based on these findings, we observed that actions in story plans often share indices with one another. In this work we explore whether this idea can help narrative planners identify solutions faster.

III. Example Domain

We will use an example narrative planning domain adapted from a short interactive narrative game [16] to illustrate our method. In the original domain there was no meaningful time index, so we modified it to include a day/night transition. The domain contains four characters—Tom, a $bandit$, a $guard$, and a $merchant$—and five locations. The $crossroads$ contains paths to each of the other four locations: Tom's $cottage$, the bandit's $camp$, the merchant's $mansion$, and the $market$, as depicted in Figure 1. Characters can only walk along these paths.

Tom needs to acquire a special potion for his grandmother. During the day, the potion can be bought for one coin from the merchant at the market, who also has a sword for sale at the same price. At night, the merchant goes to sleep at her mansion with all of her items. Tom can choose to wait for night, or the next day, at any location. Characters can steal items from others who are sleeping or dead, and characters who are armed with a sword may rob or kill unarmed characters, but these are all crimes punishable by death. The guard is armed, and he watches the market during the day, intending to punish any criminals he knows about. The bandit is a known criminal, but his whereabouts are initially unknown to the other characters. The bandit begins at his camp, where there is a chest containing one coin. Characters can take items out of the chest. The bandit is armed, and intends to acquire other valuable items such as coins and potions. Tom begins at his cottage with one coin. The story ends when Tom either succeeds in returning to his cottage with the potion, or is killed by another character.

IV. Salience Distance

We introduce a simple way to estimate the salience distance between two contiguous actions and the overall salience cost of a sequence of actions. Because we are designing fast search algorithms, we are interested in a method which can be quickly pre-computed for every pair of actions in the domain and stored in a table for fast lookup during search. The method we describe is intentionally simple; it is meant to guide a fast search, so it leaves out much of the nuance of EIM research.

Let $\pi = \{a_1, a_2, ..., a_n\}$ be a sequence of n actions. Let I be a set of salience indices. Each index $i \in I$ is a Boolean function $i(a_j, a_{j+1})$ which returns $true$ if actions a_j and a_{j+1} share that index, $false$ otherwise. We use a total of $|I| = 4$ indices in this paper, adapted from the Indexter model as follows:

- Protagonist: Two actions are linked according to this index if there exists a character that appears in the arguments of both. We do not require a predefined story protagonist, and instead generalize this index to include any character. EIM research has also generalized this dimension to represent other important entities involved in the situation, including other characters, objects, and even ideas [24].
- Time: Actions are linked if the same discrete time period appears in the arguments of both.
- Space: Actions are linked if the same location appears in the arguments of both.
- Causality: Actions are linked if a_j has an effect which also appears in the precondition of a_{j+1}. Some narrative planners support conditional effects, which means an action's effects depend on the state in which it occurs. We do not attempt to account for these conditions—that is, we treat all effect conditions as *true* when pre-computing our lookup table.
- Intentionality: We do not account for this index. Ideally, we would say that two actions are linked if they are taken in service of the same character goal; however, our metric is meant for forward planners to use during search. This means that most actions will not (yet) be explained by any character goals at the time the metric is calculated. We could potentially use goal heuristics [18] to estimate which character goals will be used to explain the actions, but this would have a polynomial time cost. For this initial version of our method, we are interested in something that can be easily pre-computed and stored in a lookup table to be retrieved in constant time. We considered several ways to pre-compute this index, but all had a significant overlap with the protagonist and causality indices to the point that they were not helpful as separate indices.

We define the salience distance between two contiguous actions a_j and a_{j+1} as follows:

$$d(a_j, a_{j+1}) = \epsilon + (1-\epsilon)\left(1 - \frac{\sum_{i \in I} \mathbb{1} i(a_j, a_{j+1})}{|I|}\right)$$

$$\text{where } 0 < \epsilon \leq 1$$

Note that $\mathbb{1}$ is the indicator function, which is 1 when the index function $i(a_j, a_{j+1})$ returns $true$ and 0 if it returns $false$. ϵ is a constant that represents the minimum distance between two actions. The remaining part of the formula ranges between 0 and $1-\epsilon$, depending on how many indices the actions share. When two actions share all indices, their distance is ϵ. When they share no indices, their distance is 1. We use a value of $\epsilon = 0.4$ in our experiments[1].

Having defined the distance between a pair of contiguous actions, we can define the salience cost of a sequence of n actions $\pi = \{a_1, a_2, ..., a_n\}$ as:

$$s\text{-}cost(\pi) = \sum_{j=1}^{n-1} d(a_j, a_{j+1})$$

Consider the example story in Figure 2, where Tom successfully walks to the market, buys the potion, and returns home. What is the salience distance between actions a_1 and a_2? The more indices they share, the closer together they are. Both actions have Tom in their arguments, so they share protagonist. Both happen on Day_1, so they share time. Both involve the $Crossroads$, so they share space. They share causality because action a_1 has the effect $location(Tom) = Crossroads$ which is also a precondition of a_2. Since they share all indices, $d(a_1, a_2)$ is ϵ, the lowest possible value.

Consider a_3 and a_4. These two actions share protagonist, time, and space, but not causality, since buy has no effect that is also a precondition of $walk$.

The overall salience cost of the solution in Figure 2 is very low for a five-step plan, $s\text{-}cost(\pi) = 1.75$.

Now consider a different solution, in which the bandit robs Tom at the crossroads (Figure 3). The first two actions share only two indices: time, because they both happen on Day_1, and space, because they both involve the $Crossroads$. Their salience distance is 0.7, higher than the pairs in the previous story. The last two actions share all four indices, since $attack$ again involves the $Bandit$, the $Crossroads$, and Day_1, and is enabled by the effect $location(Bandit) = Crossroads$ of the previous action.

The $s\text{-}cost$ of the second plan is lower, but the second plan is also shorter. If we divide the $s\text{-}cost$ by the number of actions in the plan, we see that the average salience distance between actions is 0.35 for the first plan, but 0.37 for the

[1] We ran all experiments with values of 0.1, 0.2, ..., 0.9 and found the value 0.4 gave strong results across all domains and search methods.

a_1	$walk(Tom, Cottage, Crossroads, Day_1)$	
	☑ Protagonist ☑ Time ☑ Space ☑ Causality	$d(a_1, a_2) = 0.4$
a_2	$walk(Tom, Crossroads, Market, Day_1)$	
	☑ Protagonist ☑ Time ☑ Space ☑ Causality	$d(a_2, a_3) = 0.4$
a_3	$buy(Tom, Potion, Merchant, Coin, Market, Day_1)$	
	☑ Protagonist ☑ Time ☑ Space ☒ Causality	$d(a_3, a_4) = 0.55$
a_4	$walk(Tom, Market, Crossroads, Day_1)$	
	☑ Protagonist ☑ Time ☑ Space ☑ Causality	$d(a_4, a_5) = 0.4$
a_5	$walk(Tom, Crossroads, Cottage, Day_1)$	$s\text{-}cost(\pi) = 1.75$

Fig. 2: Example solution with low average salience cost

a_1	$walk(Tom, Cottage, Crossroads, Day_1)$	
	☒ Protagonist ☑ Time ☑ Space ☒ Causality	$d(a_1, a_2) = 0.7$
a_2	$walk(Bandit, Camp, Crossroads, Day_1)$	
	☑ Protagonist ☑ Time ☑ Space ☑ Causality	$d(a_2, a_3) = 0.4$
a_3	$attack(Bandit, Tom, Crossroads, Day_1)$	$s\text{-}cost(\pi) = 1.1$

Fig. 3: Example solution with higher average salience cost

second, indicating that the progression of actions in the first plan is generally more salient than in the second. This average distance is used later in our evaluation to show that solution sequences tend to have lower salience distance between steps than non-solution sequences.

V. Salience-Based Search

We can use this distance and cost function to define three simple variations on existing forward narrative planning search techniques. A forward planner starts in the initial state with an empty plan and adds actions to the end of the plan until it is a solution[2]. The search typically maintains a priority queue of plans it could expand next, and search techniques differ based on how they decide which plans to expand.

Let $length(\pi)$ be the number of actions in a plan π. Let $h(\pi)$ be any heuristic function that estimates how many additional actions need to be added to π before it is a solution. Let $d(a_j, a_{j+1})$ and $s\text{-}cost(\pi)$ be defined as in the previous section. In this paper, we compare five search strategies:

- **Breadth-First Search (BFS)**: Expand the plan that minimizes $length(\pi)$ first.
- **Salience Breadth-First Search (SBFS)**: Expand the plan that minimizes $length(\pi) + d(a_{|\pi|-1}, a_{|\pi|})$ first (when $|\pi| > 1$). This algorithm is like BFS but uses salience as a tie-breaker; i.e. expand the shortest plan, but when plans are the same length, choose the one with lower $d(a_{|\pi|-1}, a_{|\pi|})$.
- **Salience Uniform Cost Search[3] (SUCS)**: Expand the plan that minimizes $s\text{-}cost(\pi)$ first.
- **A***: Expand the plan that minimizes $length(\pi) + h(\pi)$ first.
- **Salience A* (SA*)**: Expand the plan that minimizes $s\text{-}cost(\pi) + h(\pi)$ first.

Note that we introduced ϵ in $d(a_j, a_{j-1})$ so that a sequence of actions can never have an *s-cost* of 0, and so that it is always true that $s\text{-}cost(\pi) < s\text{-}cost(\pi + a_{n+1})$. Some search algorithms become incomplete if we allow a sequence of actions to have a 0 cost. For example, Uniform Cost Search explores lowest cost plans first, and given enough time and memory, it will eventually find a solution if one exists (i.e. it is complete). However, if it is possible to construct an arbitrarily

[2]A solution typically needs to meet two constraints: it achieves the system author's goal and any actions taken by characters must make sense for those characters. Narrative planners differ in the specifics of how they define these constraints, and we do not require any specific definition for this paper. For our experiments, we used an early prototype of our Sabre narrative planner [19], but we expect salience distance can be useful for other planners as well.

[3]This is a variant of Dijkstra's Algorithm, but whereas Dijkstra's calculates the shortest path to all nodes, Uniform Cost Search stops once it finds a solution. Uniform Cost Search is the name of this algorithm in the AI literature, but it is a frustrating name, since costs in the problem are non-uniform. The name may refer to the property that node costs in the priority queue during search tend to be uniform.

TABLE I: Comparison of salience costs between solutions and non-solutions.

Domain					Solutions			Non-Solutions		
Name	Agents	Actions	Triggers	Length	Count	Avg. d	σ	Count	Avg. d	σ
Grammalot	4	986	1,331	7	56	0.44	0.06	9,038,112	0.55	0.06
Raiders	3	39	66	8	6	0.42	0.03	81,165	0.57	0.08
Prison	3	110	54	7	128	0.49	0.08	21,076,625	0.62	0.07

TABLE II: Search algorithm performance, average of 10 runs.

Domain	Search	Time (ms)	Nodes Visited	Nodes Generated	Time/Node (ms)
Grammalot	BFS	170,674	33,900	991,433	5.0
	SBFS	117,288	26,859	705,086	4.4
	SUCS	**14,929**	**4,281**	**101,747**	3.5
	A*	246,849	11,224	445,092	22.0
	SA*	1,461,918	69,062	2,656,714	21.2
Raiders	BFS	984	2,826	32,064	0.3
	SBFS	785	2,360	25,174	0.3
	SUCS	356	1,060	11,455	0.3
	A*	**146**	**135**	**2,413**	1.1
	SA*	10,532	7,424	111,193	1.4
Prison	BFS	633,912	>750,000	>14,494,623	0.8
	SBFS	606,279	>750,000	>13,462,941	0.8
	SUCS	502,747	673,907	11,499,384	0.7
	A*	2,624,475	>750,000	>28,853,052	3.5
	SA*	**431,607**	**149,193**	**4,402,289**	2.9

long sequence of actions with a cost of 0, then UCS would no longer be complete. All five of the above techniques are complete.

While $s\text{-}cost(\pi)$ is meant to guide the search, it is not really a heuristic in the traditional sense of measuring the distance from the current state to the goal. It is a measure of the cost of the plan built so far. Hence, we do not use it as a heuristic in SA* and do not discuss whether or not it is admissible.

VI. Evaluation

We claim that using the salience cost function (instead of the number of actions) can speed up narrative planning, at least for our test domains. The three domains we use in our evaluation are described below, each with a short note about why it was chosen and some limited attempts to control for bias in how it was designed. All domains came from our previous work; in future work we hope to repeat this analysis on domains designed by other researchers.

- *Grammalot* is the domain discussed in Section III. It was originally developed for an experiment using the Camelot game engine [16], and was meant to allow a wide variety of different stories with multiple ways for characters to achieve their goals. We have modified it to include a day/night transition via the *wait* action.
- *Raiders* is the Indiana Jones-inspired domain used to evaluate the Glaive narrative planner [18]. We designate its $travel$ action to segment time in addition to space, since these actions entail significant temporal shifts in the film the domain is based on. We chose this domain because it is an established narrative planning benchmark.
- *Prison* is based on an interactive story involving prison bullies, escape, and revenge [25]. It was designed for previous EIM studies, so it already contained multiple explicit time shifts, protagonists, and character goals.

We performed two experiments. For the first, we generated every possible sequence of actions of a certain length in each domain using an early prototype of our Sabre narrative planner [19] and measured the difference in average salience cost between solutions and non-solutions[4]. Table I shows the number of agents, ground actions, and ground triggers[5] in each domain, and the maximum length of the action sequences generated for that domain. The table also shows the number of solution and non-solution sequences and their average salience distance ($\frac{s\text{-}cost(\pi)}{length(\pi)}$) according to our earlier formula. In all cases, the average salience cost of solutions is at least one standard deviation lower than the average salience cost of non-solutions. This preliminary finding supports our hypothesis that action sequences which are solutions to the problem tend to have lower salience costs.

For the second experiment, we tested how quickly each of the proposed search techniques finds its first solution to the problem. We again used our Sabre narrative planner and, when applicable, Sabre's default heuristic, which is based on Bonet and Geffner's h^+ [26]. Table II shows the time in milliseconds required by each search (average of 10 runs), along with the number of nodes visited and generated by each search. It also gives the average time spent per node visited to demonstrate the impact of measuring salience cost during search. Each search was limited to 750,000 nodes visited, so the value $> 750,000$ indicates the search exceeded its budget before

[4] A non-solution is any sequence of two or more actions that is not a solution to the problem—for Sabre, that means that either it does not achieve the author's goal or some character actions in it cannot be explained by those characters' goals. We exclude 1-action sequences because we cannot measure the *s-cost* of a plan with only one action. No problems had solutions that contained only one action.

[5] Triggers are changes that must happen when they can. They are generally used to update character beliefs based on observations. Triggers do not count as actions. See the original paper on Sabre [19] for full details.

finding a solution.

The best performing search differed by domain. In all cases, using salience as a tiebreaker improved the time and space performance of Breadth-First Search (BFS), and Salience Uniform Cost Search (SUCS) outperformed BFS on all domains. As we expected, the time spent per node visited was about the same between BFS, SBFS, and SUCS, and between A* and SA*, showing that the added overhead of calculating salience distance was minimal. As expected, the use of the polynominal time h^+ heuristic had a much greater affect on time per node.

SUCS outperformed A* in the *Grammalot* and *Prison* domains, but not in *Raiders*. Interestingly, SA* only outperformed traditional A* in the *Prison* domain. The inconsistent performance of SA* is, we suspect, due to the quality of the heuristic function which estimates the distance from the current state to the goal. We used Sabre's default heuristic, an adaptation of Bonet and Geffner's h^+ [26], which is highly accurate for the *Raiders* and *Grammalot* problems but highly inaccurate for *Prison*; for that problem it consistently and dramatically underestimates the distance to the goal. When the current-state-to-goal heuristic is highly accurate, the initial-state-to-current-state cost has less influence on the results.

Figure 4 shows the effect of ϵ (the minimum distance between actions) for each search technique. The performance of SUCS and SA* tends to improve the higher ϵ gets. SA* fails to solve *Raiders* when $\epsilon = 0.1$ and visits as few as 367 nodes when $\epsilon = 0.9$. Recall that as ϵ approaches 1 salience has less influence on the cost, so in other words, the closer SA* gets to A*, the better it does on most problems. The notable exception is SUCS on *Prison*, where low values of ϵ are better. The salience measure shines on the problem where the current-state-to-goal heuristic consistently underestimates. We feel this is a strength of our approach; if highly accurate current-state-to-goal heuristics are available, as they are for *Raiders* and *Grammalot*, improvements may not be needed in the first place. Our improvements may be most useful for problems like *Prison* where the current-state-to-goal heuristic performs poorly.

VII. Discussion and Future Work

Measuring the salience distance between contiguous actions, and the sum of these distances for whole plans, proved a simple and easy way to speed up narrative planning for some search techniques in the domains we studied. Because we pre-calculated the distances between each pair of actions and stored them in a table for fast lookup during search, the additional cost of this reasoning is negligible. Of course, this suggests a possible direction for future work. Perhaps using more sophisticated measures of salience will lead to even greater improvements. For example, we might better capture the causality dimension by looking at causal links to all past actions, rather than just the previous action.

One important note about this work is that narrative planners typically generate plot, or story, which is distinct from the discourse, or telling of the story. A discourse might leave out actions or tell them in a different order, and salience is more a property of discourse than of story. Still, many interactive narratives use a straightforward mapping from story to discourse where all actions are told in order simply because the story and discourse are being generated on demand. So this work can still be useful to narrative planners, at least when story and discourse are similar.

We can imagine at least two explanations for why salience distance is helpful during search. The first is Chekhov's Gun. It makes sense that early actions in a plan establish conditions needed by later actions. If the first action in the solution is Tom walking to the crossroads, it is probably because either Tom or the crossroads or both will be needed for something later. If we imagine how one might render the narrative on screen during gameplay, prioritizing plans with low salience cost avoids plans that jump around unnecessarily from character to character, place to place, goal to goal, etc. For this reason, we also suspect that minimizing salience distance can improve the discourse of interactive stories that tell all actions in order, though we leave an evaluation of this claim for future work.

Another possible explanation for why salience cost is effective during search is that it may tap into the domain author's original intentions for solutions. Salience can approximate how easy it is to remember or foresee a sequence of actions. We suspect, from experience, that domain authors tend to write domains with certain solutions in mind. It seems reasonable that, consciously or not, they would structure the domain to make these solutions salient. It is worth noting that the salience-based search techniques were most effective in the domains with more total solutions, suggesting that they are doing more than simply excelling at finding prototypical solutions that were specifically intended by the author.

The fact that one approach did not dominate on all problems, and the fact that higher values of ϵ were not always better is an interesting result in itself. Specifically, the fact that SUCS was consistently better than BFS, but that SA* was not consistently better than A* suggests there may be some relationship between the $s\text{-}cost$ of a plan and the ability of the heuristic to estimate how close such a plan is to being a solution. In future work, we hope to investigate this relationship. We also plan to investigate how we can integrate salience cost into the h^+ heuristic itself, or other narrative planning heuristics, such as Glaive's [18]. The more accurately a heuristic can approximate the actual solution, the faster the search will be, and improving heuristic accuracy may improve the performance of SA*.

We would also like this work to be replicated by others and tested in other domains. We determined the value of ϵ experimentally, but we would like to further investigate how this parameter affects the search and develop recommendations for how to choose it intelligently for each domain.

References

[1] R. M. Young, S. G. Ware, B. A. Cassell, and J. Robertson, "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*, vol. 37, no. 1-2, pp. 41–64, 2013.

Fig. 4: These graphs show the number of nodes visited by five search techniques before finding the first solution to each problem (average of 10 runs) for different values of ϵ. Note that 750,000 nodes was the maximum number of nodes allowed in a search and indicates that the search failed to find a solution. Breadth-First Search, Salience Breadth-First Search, and A* are not affected by ϵ so their performance is constant. The scale of the Y axis makes small values appear close to 0, but they are not 0 (see Table II).

[2] R. E. Cardona-Rivera, J. Robertson, S. G. Ware, B. Harrison, D. L. Roberts, and R. M. Young, "Foreseeing meaningful choices," in *Proceedings of the 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2014, pp. 9–15.
[3] R. Farrell, S. G. Ware, and L. J. Baker, "Manipulating narrative salience in interactive stories using indexter's pairwise event salience hypothesis," *IEEE Transactions on Games*, vol. 12, no. 1, pp. 74–85, 2020.
[4] R. A. Zwaan and G. A. Radvansky, "Situation models in language comprehension and memory," *Psychological Bulletin*, vol. 123, no. 2, p. 162, 1998.
[5] R. E. Cardona-Rivera, B. A. Cassell, S. G. Ware, and R. M. Young, "Indexter: a computational model of the event-indexing situation model for characterizing narratives," in *Proceedings of the 3rd Workshop on Computational Models of Narrative*, 2012, pp. 34–43.
[6] R. M. Young, "Notes on the use of plan structures in the creation of interactive plot," in *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, 1999, pp. 164–167.
[7] M. O. Riedl and R. M. Young, "Narrative planning: balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 39, no. 1, pp. 217–268, 2010.
[8] J. Teutenberg and J. Porteous, "Efficient intent-based narrative generation using multiple planning agents," in *Proceedings of the 2013 international conference on Autonomous Agents and Multiagent Systems*, 2013, pp. 603–610.
[9] ——, "Incorporating global and local knowledge in intentional narrative planning," in *Proceedings of the 2015 international conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1539–1546.
[10] M. Christensen, J. Nelson, and R. E. Cardona-Rivera, "Using domain compilation to add belief to narrative planners," in *Proceedings of the AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020, pp. 38–44.
[11] S. G. Ware, R. M. Young, B. Harrison, and D. L. Roberts, "A computational model of narrative conflict at the fabula level," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, vol. 6, no. 3, pp. 271–288, 2014.
[12] M. Cavazza, F. Charles, and S. J. Mead, "Character-based interactive storytelling," *IEEE Intelligent Systems special issue on AI in Interactive Entertainment*, vol. 17, no. 4, pp. 17–24, 2002.
[13] D. Pizzi and M. Cavazza, "Affective storytelling based on characters' feelings," in *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*, 2007, pp. 111–118.
[14] J. Porteous, M. Cavazza, and F. Charles, "Applying planning to interactive storytelling: Narrative control using state constraints," *ACM Transactions on Intelligent Systems and Technology*, vol. 1, no. 2, pp. 1–21, 2010.
[15] S. G. Ware and R. M. Young, "Intentionality and conflict in The Best Laid Plans interactive narrative virtual environment," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, vol. 8, no. 4, pp. 402–411, 2015.
[16] S. G. Ware, E. T. Garcia, A. Shirvani, and R. Farrell, "Multi-agent narrative experience management as story graph pruning," in *Proceedings of the 15th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 2019, pp. 87–93.
[17] T. Bylander, "The computational complexity of propositional strips planning," *Artificial Intelligence*, vol. 69, no. 1-2, pp. 165–204, 1994.
[18] S. G. Ware and R. M. Young, "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*, 2014, pp. 80–86.
[19] S. G. Ware and C. Siler, "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*, 2021, pp. 99–106, (nominated for Best Paper).
[20] R. E. Fikes and N. J. Nilsson, "STRIPS: a new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, no. 3, pp. 189–208, 1972.
[21] D. McAllester and D. Rosenblitt, "Systematic nonlinear planning," Massachusetts Institute of Technology Artificial Intelligence Laboratory, Tech. Rep., 1991.
[22] T. Trabasso and L. L. Sperry, "Causal relatedness and importance of story events," *Journal of Memory and language*, vol. 24, no. 5, pp. 595–611, 1985.
[23] C. Kives, S. G. Ware, and L. J. Baker, "Evaluating the pairwise event salience hypothesis in Indexter," in *Proceedings of the 11th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 2015, pp. 30–36.
[24] R. A. Zwaan, G. A. Radvansky, and S. N. Whitten, "Situation models and themes," *Thematics: interdisciplinary studies*, pp. 35–53, 2002.
[25] R. Farrell and S. G. Ware, "Predicting user choices in interactive narratives using indexter's pairwise event salience hypothesis," in *Proceedings of the 9th International Conference on Interactive Digital Storytelling*, 2016, pp. 147–155.
[26] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence*, vol. 129, no. 1, pp. 5–33, 2001.