A Good Story is One in a Million: Solution Density in Narrative Generation Problems

Abstract

Narrative generation systems can be classified on a spectrum from strong autonomy to strong story. The strong autonomy side treats characters as fully independent agents but may struggle to meet the author's requirements, while those on the strong story side direct character behaviors centrally but may struggle to create the illusion of character believability. In this paper, we use benchmark story generation problems as a framework to compare the spaces of stories that could be generated by prototypical strong story and strong autonomy systems. Comparing the relative solution densities of these spaces helps us quantify how common certain desirable narrative properties are. This can be informative for system designers when deciding, for instance, whether to strictly enforce all desired properties or to generate and filter from a broader class of solutions.

1 Introduction

Interactive narrative systems must balance computational efficiency, character believability, and author intent. To achieve this balance, they may use some mixture of locally-generated character behaviors and an experience manager agent for global coordination. In Riedl and Bulitko's taxonomy of narrative systems (2013), these approaches fall along a spectrum from *strong autonomy*, where each character is implemented as a fully autonomous agent, to *strong story*, where a central agent guides all character behavior.

From a designer's perspective, the ideal level of autonomy vs. centrality depends on the system's objectives. However, to our knowledge, there has not yet been a focused investigation of the tradeoffs between the two. If we want to generate a story where characters appear to act independently, yet the plot meets the author's constraints, how easily can we get such a story by running repeated simulations of fully-autonomous characters and filtering out those that violate the author's requirements? Alternatively, how easily can we get it by generating many plots that achieve the author's goal and filtering out those with unrealistic character behaviors?

Quantitatively comparing a strong autonomy approach with a strong story approach in a meaningful way is difficult, because their outputs depend on many system-specific properties like the story domain and search strategies. So rather than implementing example systems and comparing them, we instead enumerate *solution sets*, counting all possible action sequences that meet different definitions of quality. This allows us to compare approaches in a more system-agnostic way by measuring how common desirable narrative properties are (and thus how likely a system would be to find a story with those properties). For various definitions of narrative quality, we measure how densely—or rather, how sparsely—solutions are distributed throughout the search space.

We consider story generation problems that have an *author goal*, something which should be true by the end, and one or more *character goals* for each character. Character goals may synergize or conflict with those of other characters and the author. At a high level, we consider two basic kinds of stories: *structured stories* that achieve the author goal, and *stories with intentionality* where every character action can be explained by character goals.

We use the set of solutions generated by a classical planner to represent the extreme version of a strong story system, one that reasons only about structure and ignores character intentionality. On the other end of the spectrum, we use the set of solutions that could be generated by a multi-agent system to represent the extreme version of a strong autonomy approach, one that only simulates intentional character behavior and ignores the author goal.

Most story generation systems lie somewhere in the middle. We consider two kinds. The first is a narrative planner, which builds a plan to achieve the author goal out of actions that can be explained by character goals. It is a structured storyteller constrained by character intentionality. The second, inspired by a drama manager architecture, generates intentional character behavior but prevents characters from acting when they would make the author's goal impossible. It is an intentionality-based simulation constrained by structure.

In this paper, we compare the solution sets resulting from different definitions of story quality. In Section 2, we discuss how existing story-generation systems position themselves on the spectrum from strong autonomy to strong story. In Section 3, we use a common planning framework to formally define a group of solution sets, representing the possible outputs of prototypical systems that lie on different parts of this spectrum. In Section 4, we present the results of enumerating these solution sets for several benchmark story-generation domains. In Section 5, we discuss the implications of these results for story generation research and design.

2 Related Work

Since almost all narrative systems exist on the strong-story-to-strong-autonomy spectrum, we cannot cover all related work; see the surveys of Riedl and Bulitko (2013) and Kybartas and Bidarra (2016) for a more comprehensive account. Instead, we provide an overview of the spectrum with an emphasis on the models that most directly influence our experiments.

TALE-SPIN (Meehan 1977) is an early example of a strong autonomy system. The experience manager queries the author about world objects and character beliefs, relationships, and motivations (in terms of basic drives like hunger and thirst), as they become relevant; beyond this, the story plays out freely based on how these elements interact with the system's physical and social dynamics (Wardrip-Fruin 2009). In contrast, Dehn's response in the form of AUTHOR (1981) provides one of the original examples of the strong story mentality. Rather than simulating the events of the story world, Dehn proposes trying to simulate the mind of an author inventing the story through top-down "conceptual reformulation" from initial high-level goals about story structure. Many middle-of-the-spectrum approaches have used a drama manager agent to direct the story; for instance, Mateas and Stern (2002) discuss a paradigm where the drama manager occasionally imposes new goals and behaviors on otherwise-autonomous characters.

Some systems balance story structure and character believability by generating stories on one end of the spectrum and then filtering for those that meet criteria on the other end. For example, Scéalextric Simulator (Riegl and Veale 2018) is a strong autonomy system that generates many simulations. It selects stories from among the simulation traces by identifying the ones that score highest on certain story-coherence metrics. Felt (Kreminski, Dickinson, and Wardrip-Fruin 2019) also extracts stories from simulations, but with an emphasis on finding subsets of one large simulation. It maintains a database of events from a story world and provides a query language for "sifting" the world history to find event patterns that match a desired story structure.

Typical narrative planners like IPOCL (Riedl and Young 2010) and Glaive (Ware and Young 2014) are based on classical planners that are then further constrained by models of character believability. For example, Glaive performs a forward search through the space of all classical plans, allowing character actions to be unexplained when they first occur. Ideally, they will become explained by future actions, but when this does not happen, that search effort is wasted. IPOCL has a similar mechanism where the need to explain an action can be ignored in the hopes that it will be explained differently later; when it isn't, work is wasted. These systems are generating strong story plans and ignoring ones that

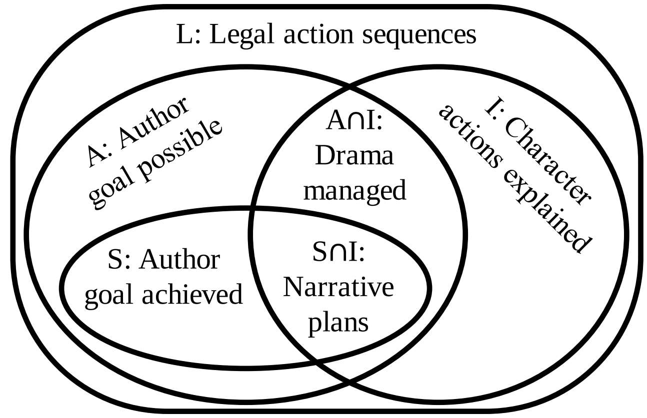

Figure 1: How different solutions spaces relate to one another

do not also meet a strong autonomy definition of quality. Teutenberg and Porteous's IMPRACTical (2013) addresses this limitation by using heuristics at every moment to determine what actions make sense for characters and then building a plan from those actions. This makes IMPRACTical less likely to waste effort, but because it relies on heuristics to decide what characters would do, it does not provide guarantees as strong as Glaive's about the explainability of actions by character goals.

Some systems use classical planners with no modifications, but rather compile story structure into the planning problem. Porteous, Cavazza, and Charles's interactive *Merchant of Venice* (2010) defines milestone events that must always occur in any solution. This approach can leverage the extensive research on fast classical planning and avoids the need for an explicit model of narrative structure or character believability to be built into the planner; however, it transfers much of the burden for creating structure and believability from the algorithm to the human author.

3 Defining Solution Sets

For several definitions of a *solution* to the narrative generation problem, our goal is count the number of stories that meet those definitions, and to compare those counts to one another and to the size of the space as whole. We do this using a common planning framework that can be constrained to generate every solution meeting various definitions. We are interested in five of the solution sets illustrated in Figure 1 and defined formally in the remainder of this section:

- L: The set of all *legal* action sequences. This represents the entire search space.
- S: The set of *structured* stories, which achieve the author goal by the end. These are the stories that could be generated by a classical planner.
- I: The set of stories with *intentionality*, where every character action can be explained by character goals. These are the solutions that could be generated by a multi-agent simulation.

- $S \cap I$: The set of stories that achieve the author's goal and are composed of explainable character actions. These are the stories that could be generated by a narrative planner.
- $A \cap I$: The set of stories composed of explainable character actions where the author's goal remains possible (even if it is never actually achieved). These are the stories that could be generated by a prototypical drama-managed architecture, where a drama-manager agent has a limited ability to constrain the actions of otherwise autonomous character agents.

The Narrative Generation Problem

Formal definitions of these spaces follow. They are based on the PDDL planning formalism (McDermott et al. 1998), with the extension proposed by Riedl and Young (2010) to distinguish between the author goal and character goals, and the extension proposed by Ware and Young (2011) to represent conflict and failed plans.

A narrative generation problem is a tuple $\langle s_0, C, A, g_a, G(c)\rangle$. s_0 is the initial state of the story world. C is a set of objects in the world representing the story's characters. A is a set of actions. Each action has a precondition and an effect, which are logical propositions. Every action also defines a (possibly empty) set of consenting characters from C. g_a is the author's goal, a logical proposition. For every character $c \in C$, $G(c)$ is a set of logical propositions that are character goals for c.

Consider an example from one of the benchmark domains used in this study (Ware et al. 2019), where a bandit steals a gold coin from another character called the player (the protagonist of the interactive story). The precondition is that the bandit and player are both alive, both at the town crossroads, and that the player has the coin. The effect is that the bandit has the coin. Though this action involves two characters, only the bandit is a consenting character, because only the bandit needs a reason to act, while the player is a victim. The author's goal for the problem is that the player has medicine (which can be bought with the coin). The bandit's goal is to have the coin, an example of character goals conflicting with author goals.

Our implementation uses a simple logical language of binary fluents (e.g. the player is either alive or not alive). It also includes special modal propositions of the form $intends(c, g)$, where c is a character and g is a character goal. When $intends(c, g)$ is true, c may act to achieve g.

Legal Stories

A story p is a sequence of n actions $\{a_1, a_2, ..., a_n\}$ that implies a sequence of $n + 1$ states $\{s_0, s_1, s_2, ..., s_n\}$ such that action a_1 occurs between states s_0 and s_1, action a_2 occurs between states s_1 and s_2, and so on.

A story p is *legal* (i.e. $p \in L$) iff s_0 is the initial state of the narrative generation problem, and for i from 1 to n, the precondition of a_i is true in state s_{i-1} and state s_i is identical to s_{i-1} except that the effect of a_i is now true. In other words, a legal sequence of actions starts from the problem's initial state, the precondition of every action is true immediately before it occurs, and the effect of each action has been imposed on the state immediately after it. The set L of legal action sequences represents all logically consistent stories that any system could tell—that is, stories that do not violate the system's internal logic of what is possible.

Structured Stories

A story p is *structured* (i.e. $p \in S$) iff it is legal and the author's goal g_a is true in the story's final state s_n. This is the definition of a valid classical plan, and it represents the stories that could be told by a strong story system that ignores character motivation. In our running example, the bandit could steal the medicine from the market and deliver it to the player, which achieves the author's goal but makes no sense given the bandit's goals.

Intentional Stories

An action a_i is *explained for* character c by character goal g in state s_{i-1} iff:

1. Starting from state s_{i-1}, there exists a legal sequence of m actions (called an explanation) that begins with action a_i and ends in a state s_m where g is true.
2. c is a consenting character for every action in the explanation.
3. For $i < m$, $intends(c, g)$ is true in state s_i.
4. The explanation does not contain a strict subsequence that also meets these criteria.

In other words, we can explain why a character took an action in terms of one of that character's goals if the character currently has the goal, can form a plan starting with that action to achieve the goal, and the plan does not contain unnecessary or redundant actions.

Note that an explanation only needs to exist; it does not actually need to occur in the story. Say the player walks to the crossroads. This action can be explained because the player could then walk to the town and buy the medicine. The bandit's robbery prevents that plan from actually occurring, but the player's action of walking to the crossroads is still explained.

An action a_i is *explained* in state s_{i-1} iff, for each of its consenting characters, there exists a goal for that character that explains that action in that state. In other words, an action is explained when it is explained for every character who needs a reason to do it. The player can buy the medicine from the merchant because the player wants the medicine and the merchant wants the coin.

A story p is intentional (i.e. $p \in I$) iff every action is explained. I is the set of stories that could be generated by a strong autonomy simulation that reasons about character motivation but ignores the author's goal.

Hybrid Stories

We also consider two solution sets that represent common approaches for addressing both structure and intentionality.

The set $S \cap I$ is all stories that achieve the author's goal using only actions that are explained by character goals. This is the set of stories that would be generated by many narrative planners.

Our model of drama management ($A \cap I$) is similar, but approaches the middle of the spectrum from the strong autonomy side. Rather than using a centralized planner to control all agents, it adds a coordinating agent to "herd cats," ensuring the cast of self-directed characters does not violate the author's goal. For this study, we represent a drama manager as an oracle that prevents a multi-agent simulation from taking an action that would make the author's goal impossible.

A story p is said to be *potentially structured* (i.e. $p \in A$) iff there exists a legal sequence of actions that begins with the actions in p and the author's goal g_a is true in that sequence's final state. In other words, set A represents all stories in which the author's goal has not yet been made impossible. For example, if the bandit kills the player, it would be impossible for the player to obtain the medicine.

The set $A \cap I$ is all stories composed of intentional character actions that do not prevent the author's goal from being achieved. Without precluding other styles of drama management, this represents one interpretation of a drama management framework.

Implementation

To bound the size of a solution space, we set a limit m on the length of a sequence. That is, we allow a story to contain at most m executed actions—events that actually occur. An explanation for a given action a may also contain up to m actions (including a itself). These explanations represent intended but not necessarily executed actions, so they may not appear in the story itself. Stories are considered the same if their executed actions are the same; i.e. the same set of executed actions with different explanations would be considered the same story.

Finally, to make a fairer comparison between more-constrained spaces like $S \cap I$ and less-constrained spaces like I, we make two assumptions to increase the likelihood of achieving the author goal. First, a story ends immediately when the author goal is achieved (to prevent it from being undone later). Second, a story will be extended as far as possible rather than terminating while there are still viable actions (assuming the author goal has not been achieved). For example, assuming $m > 1$, the 1-action story where the player walks to the crossroads is not in I (even though all its actions are explained) because there are still more explainable actions that can be taken, and the story has not yet reached its maximum length.

We implemented this story generation framework as an answer set program (Brewka, Eiter, and Truszczyński 2011). This allowed us to count the number of solutions by having the Clingo solver (Gebser et al. 2018) enumerate all answer sets corresponding to stories in each space. Our encoding was based on an existing encoding for generating classical plans (Dimopoulos et al. 2017). In order to prove a story's membership in I, it also generated explanations for actions with consenting characters; in order to prove a story's membership in A, it generated a sequence of m steps starting from the final executed step to verify that the author goal was still reachable by the end of the story. We used the Asprin extension (Brewka et al. 2015) to exclude redundant answer sets (e.g., identical stories with different explanations) and stories that terminated early when they could have been extended.

4 Experiments

Domains

We compared the solution spaces resulting from the above definitions on the following benchmark planning domains and their accompanying "canonical" problem instances:

- `aladdin` from Riedl and Young (2010) .
- `raiders` from Ware and Young (2014) .
- `storygraphs`,[1] the *Best-Laid-Plans*-inspired (Ware and Young 2015) domain used by Ware et al. (2019); we omit the belief semantics and character goal prioritizations, so that the world is fully observable and characters can pursue any of their goals at any time.
- `villains`, a domain used in PLOTSHOT (Cardona-Rivera and Li 2016); we omit the "discourse" actions used by PLOTSHOT to control story presentation and use only the "fabula" actions used to determine story contents.
- `western` from Ware (2014) .

We varied the parameter for maximum number of executed steps. For each combination of domain, maximum length, and solution space, we computed the number of unique solutions in terms of executed steps (i.e., it was possible to generate multiple different explanations for a character action, but these were not counted as separate solutions as long as the executed steps were the same).

Results

Figure 2 shows the solution-space sizes for the `raiders` benchmark. The solution spaces without intentionality, L and S, show continued exponential growth (note the logarithmic scale) as we increase the maximum length, due to having a variety of action sequences that can be repeated indefinitely with no net effect (e.g., a character travelling to a location and then back). However, among the solution spaces that have intentionality, the difference between the strong autonomy space of I and the intermediate space of $A \cap I$ is small, and the difference between these spaces and the "strong story" space of $S \cap I$ becomes smaller as the maximum sequence length increases. A similar principle applies for `villains` (Figure 3). In `aladdin` (Figure 4) and `western` (Figure 5), the large branching factor prevented us from enumerating solutions up to a maximum sequence length where $S \cap I$ solutions existed, but as before, we saw only a small difference between $A \cap I$ and I.

These results make sense for the context in which the domains were originally authored. The design process for most existing planning-based story generation domains has involved starting with a baseline plot in mind, and defining the actions and initial world state so the baseline plot will be generated (Porteous et al. 2015); this can lead to an emphasis

[1]The domain is described formally at: http://cs.uky.edu/~sgware/projects/storygraphs/readme.pdf

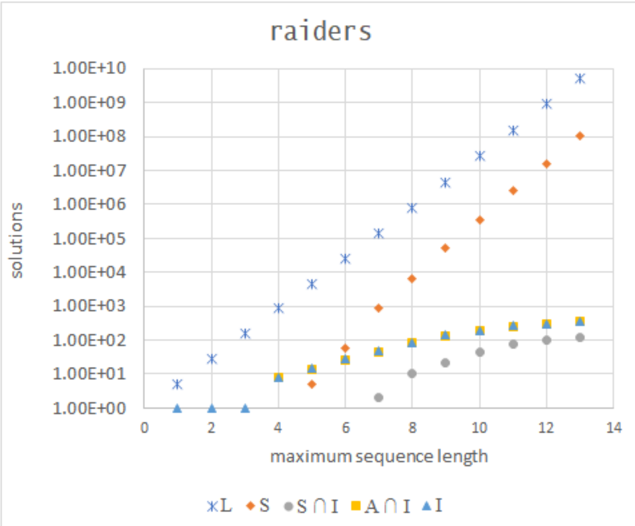

Figure 2: Results for the `raiders` domain.

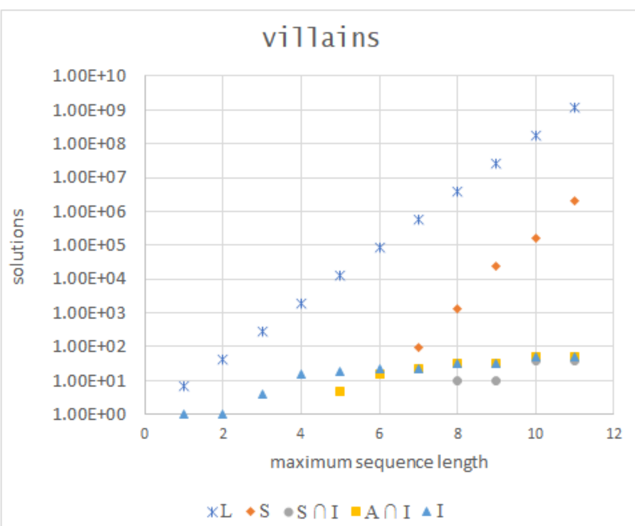

Figure 3: Results for the `villains` domain.

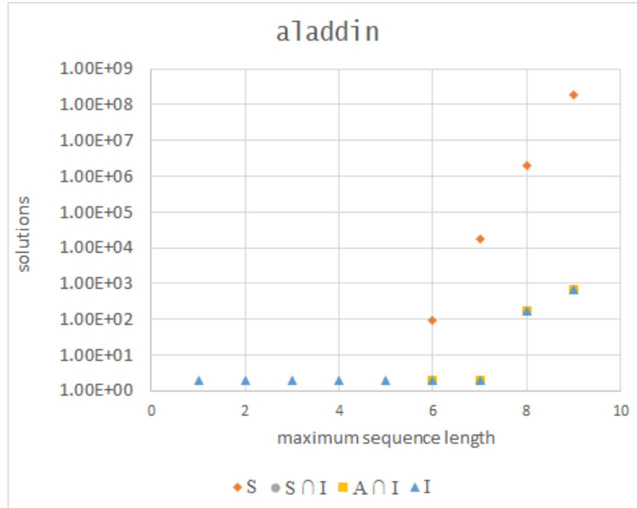

Figure 4: Results for the `aladdin` domain. L is omitted because it was too large to enumerate. $S \cap I$ markers are not visible on the logarithmic plot because the space had size 0 over the maximum sequence lengths tested.

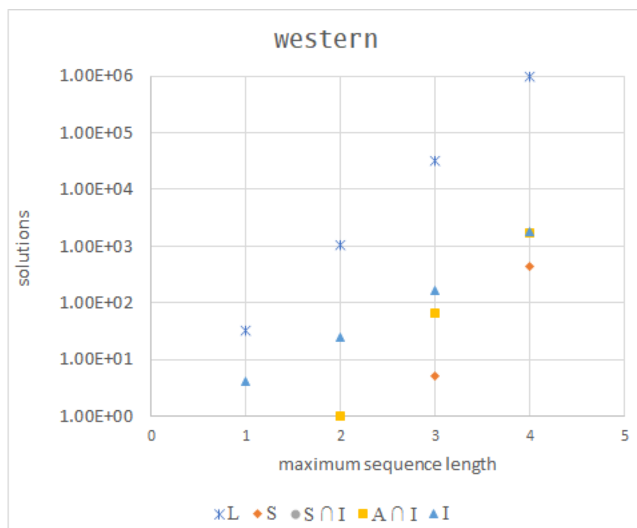

Figure 5: Results for the `western` domain. ($S \cap I$ markers are not visible because there were no solutions of these sizes.)

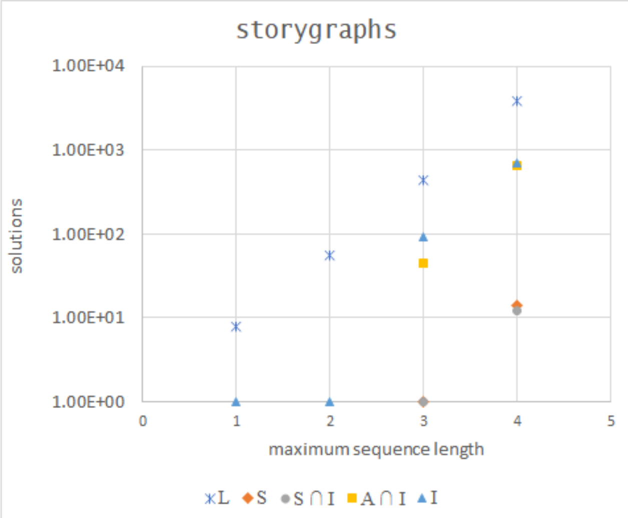

Figure 6: Results for the storygraphs domain.

on character goals that naturally align with author goals and an omission of character goals that run contrary to author goals.

In contrast to what we saw in other domains, in the sizes of storygraphs (Figure 6) we enumerated, the stronger-autonomy spaces of $A \cap I$ and I grew faster than S (though this property was shared with western), and there was a close size correspondence between $S \cap I$ and S. In other words, most of the stories that achieved the author goal also showed intentionality, but most of the stories with intentionality did not achieve the author goal; characters had a wide variety of ways to pursue their goals without necessarily furthering the author goal. Like before, the difference in space sizes reflects the design mentality of the domain; storygraphs was originally used to plan non-player character behavior in an interactive environment that emphasized giving the player character a diverse set of ways to achieve their goal or die trying.

Across the domains explored here, and across story lengths for which solutions existed, we consistently saw orders-of-magnitude differences between the space of all legal stories (L), the space of stories that achieved the author goal (S), and the space of stories where all character actions were explainable (I), though the larger space out of S and I differed by domain. When narrative plans existed ($S \cap I$), they tended to consitute the majority of either S or I, whichever was smaller. Finally, our drama-manager-inspired solution space $A \cap I$ constrained the set of stories only slightly compared to I, likely a consequence of the benchmark domains having few actions that could permanently jeopardize author goal achievement and few character goals that would motivate those actions; this does not cast doubt on the efficacy of drama management in general, only highlights the challenge in designing effective domain-independent criteria for drama management.

5 Conclusions

In this study, we compared the spaces of stories in a plan-based story generation context that can come about when we change various assumptions about how the characters can behave and how strongly an author goal is enforced. Classical planning can be used to generate stories where only the author goal is considered. Narrative planning combines strict enforcement of author goal achievement with character behavior constraints that promote believability. Keeping the character behavior constraints and relaxing or removing the author goal requirement gives us solution sets that model what stronger-autonomy systems might produce.

Many existing benchmarks from narrative planning were constructed to yield a specific story or a small set of similar stories, and in some of these cases, enforcing character intentionality can incidentally lead to author goal achievement much of the time. This has implications for how narrative planners are designed and tested. For instance, consider a state-space planner based on firstly determining which actions are explainable and searching an author-goal-achieving state using only those (Teutenberg and Porteous 2013), compared to an author-goal-first planner that finds the action explanations after the overall executed sequence (Ware and Young 2014). We would expect the explanation-first planner to perform exceptionally well on problems like the raiders benchmark, because chaining together a series of explainable character actions will often lead to the author goal. Conversely, consider storygraphs, where the trend was reversed — stories made of explainable character actions were much more common than stories that achieved the author goal, but most stories that achieved the author goal were also made of explainable character actions. We would expect an author-goal-first planner to have an advantage in this domain because once an author-goal-achieving plan is found, it will often be possible to explain all of the character actions afterward.

For designers of domain-specific narrative systems, investigating the density of different solution types in the domain can thus provide insight into story-generation strategies that could be effective — e.g., whether it is necessary to enforce all desired constraints, or whether it is feasible to relax some of the constraints and still get the desired properties. In the examples we saw, enforcing certain constraints explicitly (e.g., going from L to S or I) could vastly narrow down the solution space, while other constraints (e.g., going from I to $I \cap A$) often had diminishing returns that, in a real-time application, could translate to unnecessary computational costs.

Meanwhile, for designers of narrative planners where general efficiency is desired, in order to ensure robust testing, a benchmark suite should include both domains where it is hard to find author-goal-achieving stories among the stories with explainable character behavior, and the reverse; currently, there is a particular lack of benchmarks where neither feature is strongly correlated with the other, which is likely to be the case in many realistic interactive-narrative applications where we would like to deploy the planning technology.

References

Brewka, G.; Delgrande, J.; Romero, J.; and Schaub, T. 2015. asprin: Customizing answer set preferences without a headache. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Brewka, G.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54(12):92–103.

Cardona-Rivera, R. E., and Li, B. 2016. PLOTSHOT: Generating discourse-constrained stories around photos. In *12th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Dehn, N. 1981. Story generation after TALE-SPIN. In *7th International Joint Conference on Artificial Intelligence*, volume 81, 16–18.

Dimopoulos, Y.; Gebser, M.; Lühne, P.; Romero, J.; and Schaub, T. 2017. plasp 3: Towards effective asp planning. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, 286–300. Springer.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Lühne, P.; Obermeier, P.; Ostrowski, M.; Romero, J.; Schaub, T.; Schellhorn, S.; and Wanko, P. 2018. The potsdam answer set solving collection 5.0. *Künstliche Intelligenz* 32(2-3).

Kreminski, M.; Dickinson, M.; and Wardrip-Fruin, N. 2019. Felt: A simple story sifter. In *12th International Conference on Interactive Digital Storytelling*, 267–281. Springer.

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

Mateas, M., and Stern, A. 2002. Towards integrating plot and character for interactive drama. In *Socially Intelligent Agents*. Springer. 221–228.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL — the Planning Domain Definition Language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

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

Porteous, J.; Lindsay, A.; Read, J.; Truran, M.; and Cavazza, M. 2015. Automated extension of narrative planning domains with antonymic operators. In *2015 International Conference on Autonomous Agents and Multiagent Systems*, 1547–1555.

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 Bulitko, V. 2013. Interactive narrative: an intelligent systems approach. *AI Magazine* 34(1):67–77.

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

Riegl, S., and Veale, T. 2018. Live, die, evaluate, repeat: Do-over simulation in the generation of coherent episodic stories. In *International Conference on Computational Creativity*, 80–87.

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

Wardrip-Fruin, N. 2009. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. MIT Press.

Ware, S. G., and Young, R. M. 2011. CPOCL: a narrative planner supporting conflict. In *7th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 97–102.

Ware, S. G., and Young, R. M. 2014. Glaive: a state-space narrative planner supporting intentionality and conflict. In *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.; Garcia, E. T.; Shirvani, A.; and Farrell, R. 2019. Multi-agent narrative experience management as story graph pruning. In *15th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 87–93.

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