A Monte Carlo Tree Search Based Approach to Producing Stories with Excitement Curves

Thomas Hvizdos
Undergraduate Senior Thesis
Department of Computer Science
University of Minnesota
Email: hvizd007@umn.edu

Abstract

This paper presents a Monte Carlo Tree Search (MCTS) algorithm that generates stories with rising and falling excitement. Our approach is based on the well-established Freytag Pyramid model of story structure, and attempts to fit stories to a defined excitement curve. The initial implementation was supported by a user study, which asked humans to create stories in the domain our algorithm worked in. The results of that study led to a refinement of our algorithm and a data-based excitement curve. We generated stories in two domains, and observed positive effects in at least one.

1 Introduction

Story generation is a challenging problem, with a number of potential applications. Computer generated narratives can be used to create more engaging interactive experiences [1], [2], train medical professionals [3], or help people work through challenging life events [4], [5]. Unfortunately, generating stories that are sophisticated enough to be useful has proved to be an exceedingly difficult problem. Many aspects of "good" stories are not well-defined, and so creating an algorithm that produces satisfying narratives can be extremely challenging. Nevertheless, many attempts to solve the problem have been made.

One way to describe the quality of a generated story is to create heuristics based on established literary theory. Those well-defined heuristics can then be used in the generation domain as an evaluation method. This paper draws inspiration from the Freytag Pyramid [6] model of dramatic structure to create a graph that defines a desired excitement value for each action in a generated story. This "excitement curve" is embedded into an existing planning algorithm, with the goal of creating stories that have a sense of rising and falling action.

2 Related Work: Excitement, Tension Curves, and Rising Action

Research on computer generated narrative is wide-ranging and diverse. Attempts have been made to generate stories by focusing on "dilemmas", [7] learning data from existing texts [8], modelling the human creative process [9], drawing "analogies" to existing stories [10], generating stories around certain themes [11], and more. As the primary focus of this paper is on dramatic arcs and story generation, the works especially highlighted here will focus on those topics specifically.

The idea that stories should have some sort of dramatic arc is one of the oldest in history. Aristotle describes it in his *Poetics*[6], and the idea has persisted in many forms ever since, adapting in various ways to suit the dominant narrative forms of the time. In the 1800s, Gustav Freytag formulated a model for describing the arc of 5-act plays[6] whose basic structure has persisted into modern times. His discussion of exposition, rising action, climax, falling action, and denouement provide a simple, but useful, way to understand the structure of narratives.

Given the importance of dramatic arcs to human storytelling, it is unsurprising that scholars involved in the creation of computer generated narrative would attempt to use these concepts to guide their work. Leon and Gervás define several different ways to evaluate stories based on curves of any type, and specifically mention the utility of such an approach with regards to mathematically defining dramatic arcs [12]. Szilas and Richie attempt to create a framework that creates a tension arc based on paradoxes [4]. Finally, Dominguez et al. and Barros et al. extend their respective story planning algorithms to generate stories that conform to dramatic arcs, with positive results [1], [13].

Explicitly defining a tension arc is challenging, and each paper takes a different approach. However, there seems to be evidence that incorporating the idea of a tension curve or dramatic arc into narrative generation can have a positive impact.

3 A Monte Carlo Tree Search based Narrative Generator

This paper extends a narrative generation algorithm proposed by Kartal et al. [14]. Their approach focuses on creating believable stories that achieve user-defined goals, but does not consider a generated story's overall structure. This section summarizes their work to provide context for my thesis.

This work was completed under the supervision of Dr. Stephen J. Guy and with the assistance of Bilal Kartal

Defining a Story Domain

To create narratives with MCTS, a domain space must be clearly defined. Kartal et al. define stories as consisting of 4 main components: *Actors, Actions, Locations, and Items.*

- **Actors** are the characters of the story, such as John, Police, or Holmes. They have a handful of variables associated with them that keep track of their state (current location, alive/dead, angry, etc.).
- **Actions** are the events of the story, such as a character moving from one place to another, a natural disaster striking, or one character killing another. They can have restrictions on their ability to be performed depending on the story state (e.g., a character cannot move to the location they are currently at.)
- **Items** enable the performance of some actions. For example, a kill action requires the killing actor to have a murder weapon, and a steal action requires the victim to be holding an item that can be stolen.
- **Locations** are simply places that actors or items can be.

By defining specific instances of these four factors, one can construct a specific story domain (e.g., Detective or Fairy Tale) and an initial story state. A story is then represented by a series of states, connected by the *Actions* that transition between them.

Evaluation Heuristics: Goals and Believability

It would be desirable to have a metric to separate good stories from poor ones, especially because the MCTS algorithm requires some way of evaluating stories to function. Kartal et al. do this by combining two scores: percentage of goals achieved, and believability.

- **Goals Achieved:** Each story created has a number of defined objectives that should occur at some point in the story. A story receives a score between 0 and 1 depending on how many of these goals it achieves. (e.g. a story that satisfies 1/4 of its goals would score .25 on this metric.)
- **Believability:** This metric attempts to model the reasonableness of a given action in context. Base believability values are creator defined, but can change slightly depending on the story state. (e.g. an actor performing a kill action is more believable if that actor is angry.) The believability of a story as a whole is the product of all of its actions' individual believability values.

Believability and *Goals Achieved* are multiplied together to get the story's total score.

Introducing Monte Carlo Tree Search

With well-defined story domains and evaluation methods, it is possible to use MCTS to generate stories that maximize the evaluation heuristics. The possibility space of the domain is modelled as a tree, with story states as nodes and *Actions* as edges from those nodes. The initial story state becomes the root of the tree, and generating a good story can be formulated as a search problem in a large domain.

MCTS is an anytime algorithm which is well suited to this sort of search problem. It uses random sampling to build a tree that explores the search space selectively, biasing toward outcomes it estimates will produce the best results. A more in-depth discussion of the algorithm and potential applications can be found at [15], but the basic structure is as follows:

1. **Selection:** The algorithm moves to the tree node that is the most promising at the moment. However, since intermediate story state scores are estimates rather than exact results, it is necessary to select new paths in addition to paths that are currently highly rated. This is referred to as the ”exploration vs. exploitation” dilemma. One well-established approach to balance these two goals is the use of Upper Confidence Bounds[16], which is what Kartal et al. do.
2. **Expansion:** If possible, create a new node extending from the last selected node, representing a new action and resulting story state.
3. **Rollout:** Simulate random actions from the expanded node until the story reaches a terminal state. Use the score of that story to calculate the value of the current node. This step is the key to MCTS, as evaluating this random final result gives us a rough estimate of how useful taking that step would be while saving computation.
4. **Back-propagate:** Back-propagate that score to all the parent nodes, and return to step 1 until the algorithm ends. When it does, return the path with the highest current score.

Kartal et al. introduced two search balancing techniques to the algorithm to accelerate computation and improve the resulting stories. We refer the reader to the original paper for complete details [14].

Sample Story

Although relatively straightforward compared to some other story generation approaches, the approach described in Kartal et al. is able to generate compelling results. A sample story is given in **Fig. 1**, which had the goal of two people dead, and the murderer arrested.

4 Adding Excitement

Other narrative generators have biased their stories towards conforming to specific dramatic arcs [1], [4], [13]. The success of those approaches inspired our attempt to bring similar ideas to the MCTS generation algorithm proposed in [14]. We refer to this metric as ”Excitement.”

Recall that the base algorithm computed a story's score with the equation:

$$E(A) = G(A)B(A)$$

where A is a given story, E(A) is the total score of that story, and G(A) and B(A) are the Goals Achieved and Believability heuristics. Adding a third Excitement criteria X(A) to that equation gives us

$$\hat{E}(A) = E(A)X(A)$$

Of course, the concept of “tension” or “excitement” is not well-defined, and so some model must be chosen to provide

Alice picked up a vase from her house. Bob picked up a rifle from his house. Bob went to Alices house. While there, greed got the better of him and Bob stole Alices vase! This made Alice furious. Alice pilfered Bobs vase! This made Bob furious. Bob slayed Alice with a rifle! Bob fled to downtown. Bob executed Inspector Lestrade with a rifle! Charlie took a baseball bat from Bobs house. Sherlock went to Alices house. Sherlock searched Alices house and found a clue about the recent crime. Bob fled to Alices house. Sherlock wrestled the rifle from Bob! This made Bob furious. Sherlock performed a citizens arrest of Bob with his rifle and took Bob to jail.

Figure 1: A sample story generated the approach proposed in [14]. This story, although strong, could be paced slightly better. There is a lull after Bob kills Alice and the Inspector that disrupts the rhythm of the action, and the conclusion is somewhat abrupt. $E(A) = 0.68$

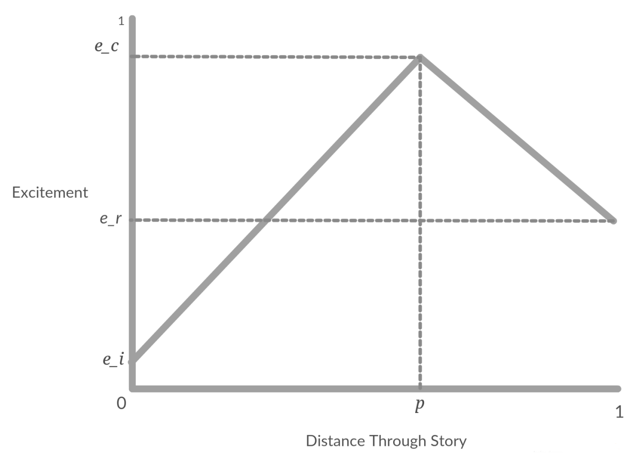

Figure 2: Abstract representation of the Excitement Curve model. e_i**,** e_c**,** e_r **and** p **can be anywhere between 0 and 1.**

X(A). We start by defining a target excitement curve. Excitement begins at an initial value (e_i), and rises to a climax excitement value (e_c) some proportion of the way through the story (p). Excitement then falls to a resolution excitement value (e_r) at the end of the story. **Fig. 2** summarizes this in graphical form.

With this curve defined, the ith action a_i in a story of length l has a desired excitement value given by

$$\begin{cases} e_i + i\frac{e_c - e_i}{l*p}, & \text{if } i < l*p. \\ e_c + (i - l*p)\frac{e_c - e_r}{i*p - l}, & \text{otherwise.} \end{cases} \quad (1)$$

To make use of this curve, we assign each action an excitement value between 0 and 1, with more exciting actions having higher values. Our defined excitement values are given in **Table 1**. We compute X(A) as the average difference between the desired excitement of an action (e_{di}), and actual excitement of an action (e_{ai}), subtracted from 1. To account for the fact that it is very unlikely for an actions desired

Action	Excitement Value
Move	0.2
Play	0.2
Wander	0.2
Pick up Item	0.4
Summon	0.4
Hide	0.5
Search for clue	0.5
Share Clue	0.5
Cry	0.6
Marry	0.7
Arrest	0.8
Fight	0.8
Love	0.8
Steal	0.8
Kill	1
Disaster	1

Table 1: Excitement values for the actions possible in our domains. These values were chosen by the author.

excitement to perfectly match the actual excitement, differences under a small threshold are ignored (this paper uses .1).

$$X(A) = 1 - \frac{\sum_{i=0}^{l} |e_{ai} - e_{di}|}{l} \quad (2)$$

5 User Study

Initially, e_i, e_c, e_r, and p were somewhat arbitrarily defined as 0, 1, .5, and .6. A survey was conducted to attempt to infer those values from human authored stories. Following an approach similar to the one used in [17], participants in the survey were asked to use our domain model to generate a typical story in which one person died, and one person was arrested. The story space was limited to actions, locations, items, and actors present in our domain. Other than those two restrictions, participants were unconstrained. It was hoped that analyzing the excitement curves of human created stories would allow us to infer useful values for e_i, e_c, e_r, and p.

In total, 17 stories were collected. Because the stories were of differing lengths, each story's excitement curve was compressed to a standard length, so that analysis on the aggregate could be done more easily. Using these normalized stories, we produced a graph (**Fig. 3**) that displayed the average excitement value at a given point in the story, along with the standard deviation.

Fig. 3 did not provide useful information to infer excitement curve parameters. Excitement remains basically flat throughout, with a high standard deviation. However, it illuminated a potential flaw in the excitement curve model. In this iteration of the model, movement actions in the story were always considered to be of low excitement. But, move actions were distributed somewhat uniformly throughout user studies, as characters often needed to change locations to advance the plot. This resulted in flat, noisy excitement curves, as actions with high excitement were negated

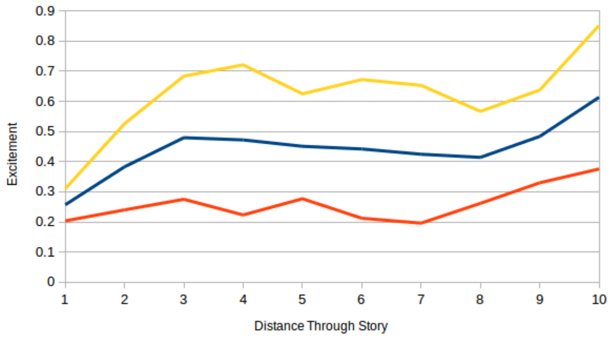

Figure 3: The initial graph for average user story excitement values. The middle curve is the average story, and the lines above and below it are +/- the standard deviation. The average is relatively flat, and standard deviation is high.

Variables	e_i	e_c	e_r	p
Conjectural	0.0	1.0	0.5	0.6
Empirical	0.55	0.4	0.9	0.7

Table 2: Excitement values for e_i, e_c, e_r, p; "Conjectural" values are those defined by the author, and "Empirical" are those learned from survey data.

by surrounding move actions. Additionally, stories that were similar (save for when characters changed locations) appeared very different. This seemed to imply that move actions should not contribute to the excitement curve, an insight that we incorporated into our model.

This also allowed us to look at the data from a slightly different perspective. We normalized the survey stories again, this time ignoring move actions. The new average participant-created story was then graphed, along with the 90th and 10th percentiles. This graph is shown in **Fig. 4**.

This is slightly more interesting. First, there seems to be a spike about 2/3rds of the way through the story, indicating many exciting events occurred around that point. This provides some support for the initial choice of .6 for p. Second, although it does not fit our initial excitement curve model, the average curve seems potentially useful for story generation, as it has two relatively steady motions.

6 Generated Stories

In the end, stories were generated using several different approaches, in two domains. Stories were generated that did or did not count move actions for excitement, and used two different sets of values for e_i, e_c, e_r, and p, shown in **Table 2**.

For comparison, stories using the original MCTS algorithm (without excitement) were also generated. Actors, Items and Locations were present in smaller numbers in these generated stories than they were in the original paper due to hardware constraints. Eight representative stories are presented (**Fig. 4-11**), along with their $\hat{E}(A)$ score:

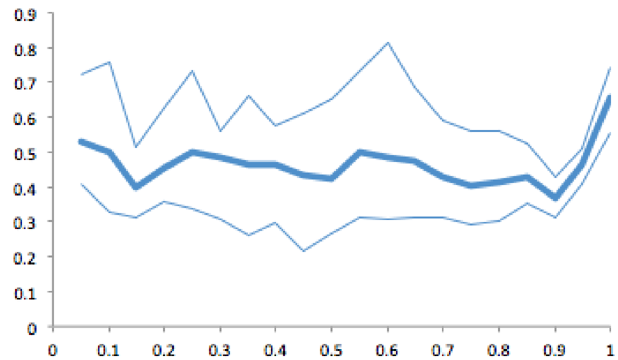

Figure 4: The graph obtained from recalculating the average user story while discounting move actions. The middle curve is the average story, and the lines above and below are the 90th and 10th percentile values at that point. The average curve seems more structured here, and there is an excitement spike at .6 of the way through the story, which supports our initial choice of setting $p = 0.6$

Princess picked up the treasure at the castle. King went to the house. Prince picked up the sword at the castle. Prince stole Princess's treasure. Jack stole Prince's treasure. Prince killed Jack with the sword at the castle. Princess witnessed the crime. Prince and Princess fell in love with each other. Prince and Princess got married!

Figure 5: Generated story in the fairy tale domain using the original algorithm. $\hat{E}(A) = 0.59$

Princess went to the house. Prince picked up the sword at the castle. King and Prince fell in love with each other. Prince picked up the treasure at the castle. Jack stole Prince's treasure. Prince killed Jack with the sword at the castle. King witnessed the crime. King and prince got married!

Figure 6: Generated story in the fairy tale domain using conjectural values, and counting move actions towards the excitement curve. $\hat{E}(A) = 0.47$

King picked up the treasure at the castle. Jack went to the house. King picked up the sword at the castle. King and prince fell in love with each other. Prince summoned princess. King and Prince got married! Princess stole king's treasure. King killed princess with the sword at the castle. Prince witnessed the crime.

Figure 7: Generated story in the fairy tale domain using conjectural values. Move actions do not count towards the excitement curve. $\hat{E}(A) = 0.42$

Prince and Princess fell in love with each other. Jack picked up the treasure at the castle. Prince stole Jack's treasure. Jack hid at castle. Prince and Princess got married! Jack picked up the sword at the castle. Jack killed Prince with the sword at the castle. King witnessed the crime. Princess witnessed the crime.

Figure 8: Generated story in the fairy tale domain using empirical values. Move actions do not count towards the excitement curve. $\hat{E}(A) = 0.46$

Holmes went to the station on the way to the office. John picked up the knife at the office. John killed Holmes with the knife at the office. Police witnessed the crime. Police picked up the revolver at the office. Police arrested John with the revolver at the office.

Figure 9: Generated story in the detective domain using the original algorithm. $\hat{E}(A) = 0.02$

John went to the village on the way to the office. John picked up the money at the office. Police went to the village. Holmes picked up the game at the house. John picked up the knife at the office. Police picked up the revolver at the village. Holmes went to the village on the way to the office. When suddenly, an earthquake struck house! But luckily, there was nobody there. Holmes stole John's money. John killed Holmes with the knife at the office. John went to the station. Police went to the station on the way to the street. Police went to the station. Police arrested John with the revolver at the station.

Figure 10: Generated story in the detective domain using conjectural values, and counting move actions toward the excitement curve. $\hat{E}(A) = 0.03$

Police went to the station. John picked up the money at the office. Police picked up the revolver at the station. Holmes picked up the game at the house. John picked up the knife at the office. Holmes went to the street on the way to the village. John went to the village. Holmes stole John's money. John got his money back from Holmes. John killed Holmes with the knife at the village. When suddenly, an earthquake struck street! But luckily, there was nobody there. John and Holmes played together with game. John went to the station. Police arrested John with the revolver at the station.

Figure 11: Generated story in the detective domain using conjectural values. Move actions do not count towards the excitement curve. $\hat{E}(A) = 0.06$

Police went to the village. Police picked up the revolver at the village. John went to the station. Holmes went to the office. Police went to the house. Holmes picked up knife and money at the office. Police went to the office. John went to the office. John stole Holmes's money. Holmes killed John with the knife at the office Police witnessed the crime Police arrested Holmes with the revolver at the office.

Figure 12: Generated story in the detective domain using empirical values. Move actions do not count towards the excitement curve. $\hat{E}(A) = 0.27$

7 Interpretation

These stories were not formally evaluated, making it difficult to make strong claims about the effects of the various ways we incorporated excitement. In general, though, adding the excitement curve produced noticeable changes in the detective domain, with characters becoming more mobile, stories becoming longer, and steal actions becoming more likely. However, the fairy tale domain was not noticeably affected by the introduction of the Excitement heuristic. In general, discounting move actions and using the curve from the human-generated stories did not have a noticeably large impact on stories generated in either domain. Using the empirical curve did increase $\hat{E}(A)$ values in the detective domain, but stories looked similar to those generated using the conjectural curve. This may be an indication that using the user curve produces subtle benefits that are not immediately obvious.

There are several factors that I believe contributed to these results. First, the Goals Achieved heuristic has a disproportionately large impact on the stories that are generated. Our domains had relatively small numbers of goals, which meant each goal became very important. For example, in the detective domain we used, there were only two goals defined, meaning that a story that only achieved one goal would automatically have a maximum possible value of only .5, even if everything else was perfect. This strongly incentivizes completion of all goals, and helps explain why changing the target excitement curve had little impact. The user excitement curve penalized very exciting actions like killing or falling in love, but if those actions were designated as goals, the benefit of including them greatly outweighed the cost. It may be possible to easily resolve this issue in future work by reducing the weight of the Goals Achieved heuristic.

Second, the original algorithm already generated stories with dramatic arcs, to a certain extent. We generally want high excitement actions like killing to occur towards the end of a story. However, many actions with high excitement values already occur towards the end of stories, because they have prerequisite actions that must occur before they are believable or possible. For instance, killing has a higher believability if someone is angry, which requires that they have been stolen from, which requires them to have picked up an item. Our excitement metric may enforce this structure more explicitly, but the original stories were often close enough to a satisfying dramatic arc that the change was not significant.

We could test this by redefining Actions to be possible with less restrictive conditions (e.g., killing no longer requires an item), removing contextual believability modifiers, and then re-evaluating stories with and without excitement curves.

8 Limitations and Future Work

First, our user study was somewhat small. Only having 17 stories makes extracting useful aggregate data somewhat challenging. Had the survey sample been larger, it might have been possible to learn more useful e_i, e_c, e_r, and p values, or see stories clustering around certain structures.

Perhaps most significantly, there was no formal evaluation of generated stories. While critical to the MCTS algorithm, the story score value does not necessarily measure how "good" a story is to a human reader. Some results were observable simply by inspecting generated stories, but those judgements are extremely subjective. It would be much better to define a few axes to measure stories on, create a survey that asks participants to rate generated stories on those axes, and see if more data-driven conclusions could be drawn from the results. These results would be less subjective, and might allow for the observation of subtle differences between stories.

Third, excitement values were chosen somewhat arbitrarily by the author. Again, it would be preferable to learn these values by conducting some sort of survey.

Finally, as mentioned above, domains were somewhat small to allow MCTS to run in reasonable time on the available hardware. Enlarging the domain space may have caused the extra structure provided by the Excitement metric to be more significant.

9 Conclusions

Adding the excitement curve metric improved stories in the detective domain by increasing the prevalence of steal actions, making those stories more believable. The fairy tale domain was not obviously affected by the addition of the Excitement heuristic, but it is possible that more rigorous evaluation of the resulting stories would reveal subtle differences. Any lack of obvious change is likely attributable to the fact that the existing heuristics already provided rising and falling action to some degree, a sign that domain heuristics can implicitly provide the benefits of good dramatic arcs if they are well-designed.

Acknowledgements: Thanks to Dr. Stephen J. Guy and Bilal Kartal for advice and assistance throughout the research process, and Dr. Stephen J. Guy, Bilal Kartal, and Rahul Narain for reviewing and commenting on this paper.

References

[1] C. A. Dominguez, Y. Ichimura, and M. Kapadia, "Automated interactive narrative synthesis using dramatic theory," in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, 2015, pp. 103–112.

[2] J. O. Ryan, A. Summerville, M. Mateas, and N. Wardrip-Fruin, "Toward characters who observe, tell, misremember, and lie," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[3] A. Lindsay, F. Charles, J. Read, J. Porteous, M. Cavazza, and G. Georg, "Generation of non-compliant behaviour in virtual medical narratives," in *Intelligent Virtual Agents*, 2015, pp. 216–228.

[4] N. Szilas and U. Richle, "Towards a computational model of dramatic tension," in *2013 Workshop on Computational Models of Narrative*, 2013, pp. 257–276.

[5] N. Habonneau, U. Richle, N. Szilas, and J. E. Dumas, "Interactive storytelling: 5th international conference, icids 2012, san sebastián, spain, november 12-15, 2012. proceedings," in, D. Oyarzun, F. Peinado, R. M. Young, A. Elizalde, and G. Méndez, Eds. 2012, ch. 3D Simulated Interactive Drama for Teenagers Coping with a Traumatic Brain Injury in a Parent, pp. 174–182.

[6] G. Freytag and E. J. MacEwan, *Freytag's Technique of the drama : An exposition of dramatic composition and art. An authorized translation from the 6th German ed.* Chicago : Scott, Foresman, 1900.

[7] H. Barber and D. Kudenko, "Generation of dilemma-based interactive narratives with a changeable story goal," in *Proceedings of the 2Nd International Conference on INtelligent TEchnologies for Interactive enterTAINment*, 2007, 6:1–6:10.

[8] N. McIntyre and M. Lapata, "Plot induction and evolutionary search for story generation," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 1562–1572.

[9] R. P. y Pérez and M. Sharples, "Mexica: A computer model of a cognitive account of creative writing," pp. 119–139, 2001.

[10] S. Ontanon and J. Zhu, *The sam algorithm for analogy-based story generation*, 2011.

[11] C. Hargood, D. Millard, and M. Weal, "A thematic approach to emerging narrative structure," 2008.

[12] C. León and P. Gervás, "Prototyping the use of plot curves to guide story generation," in *Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012)*, 2012.

[13] L. M. Barros and S. R. Musse, "Towards consistency in interactive storytelling: Tension arcs and dead-ends," *Comput. Entertain.*, 43:1–43:17,

[14] B. Kartal, J. Koenig, and S. J. Guy, "User-driven narrative variation in large story domains using monte carlo tree search," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems*, 2014, pp. 69–76.

[15] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *Computational Intelligence and AI in Games, IEEE Transactions on*, pp. 1–43, 2012.

[16] L. Kocsis and C. Szepesvri, "Bandit based monte-carlo planning," in *In: ECML-06. Number 4212 in LNCS*, 2006, pp. 282–293.

[17] B. Li, S. Lee-Urban, G. Johnston, and M. Riedl, "Story generation with crowdsourced plot graphs," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.