Pareto-Based Narrative Planning: Making NPCs More Rational

Molly Siler, Mira Fisher, Stephen G. Ware

Narrative Intelligence Lab, Department of Computer Science, University of Kentucky
329 Rose Street
Lexington, Kentucky 40506
molly.siler@uky.edu, mira.fisher@uky.edu, sgware@cs.uky.edu

Abstract

A narrative planner decides the actions for all characters in a story while justifying each action according to the acting characters' own individual goals. However, an action that contributes to a goal may still seem irrational when considered alongside other actions the character may take; for instance, it may sacrifice a more important goal, or expose the character to unnecessary risks. We redefine the narrative planning character model to use a multiobjective framework where character actions are chosen from a Pareto front of *best* and *safest* options. We discuss how this framework can be applied to generate a policy for how the non-player characters should behave in any given state of an interactive narrative, and how we applied such a policy in the *Traffic Stop* de-escalation training simulation.

1 Introduction

Narrative planning (Riedl and Young 2010) is a centralized approach to story generation that adopts some of the strengths of multiagent-system-based generation. Story characters have goals, but they are not autonomous agents. Instead, the planner manipulates the characters; however, the planner may choose only from actions that the characters themselves would want to take, according to some model of character decision-making.

One ongoing area of narrative planning research is defining the character model. Existing narrative planners (see Section 2) are effective at deciding whether a character action makes sense in a vacuum, namely, whether the action individually could be part of a plan to achieve the character's goals. However, there has been less work on considering the action in the context of the character's other options. A character's plan to pursue one goal may become less rational when there is an equally viable way to pursue a more important goal, or when the plan would achieve the goal but result in severe negative consequences for the character.

This paper proposes a narrative planning model that improves on previous versions by evaluating character actions in the context of other available actions. Characters reason about possible futures in a manner inspired by game tree search. They recursively anticipate others' actions and try to get closer to outcomes good for them and to eliminate the possibility of outcomes bad for them. There is sometimes a tradeoff between getting closer to a better outcome and preventing a worse outcome; for instance, guaranteeing safety from a threat may require that a character abandon their most valued goals. We treat pursuing rewards and avoiding risks as separate objectives of a multiobjective problem, and we use the notion of a Pareto front over these two objectives to determine which actions a character considers.

We also discuss the application of our model to *Traffic Stop*, an interactive narrative application introduced in our previous work (Fisher, Siler, and Ware 2022). As a simulation of a police de-escalation situation, it features NPCs who require multi-layered reasoning about threats and opportunities from each other and from the player; our model has helped to define a space of NPC behaviors appropriate to the system's pedagogical goals.

2 Related Work

Alhussain and Azmi (2021) provide a broad survey of story generation methods. Here we focus on the specific thread of narrative planning research that has been defining the space of valid stories by broadening and narrowing the requirements that an individual character's plans must meet.

Young (1999) proposed the use of planning algorithms for story generation because they offer a formal, generative model of a series of temporally- and causally-linked events. Although planning has been present since the beginnings of story generation research (Meehan 1975; Lebowitz 1985), Young noted that early systems largely used *ad hoc* models of plans; Young instead adopted STRIPS-style (Fikes and Nilsson 1972) classical planners as a new focal area. It soon became clear that the space of all valid plans (region C in Figure 1) was not precise enough as a definition of acceptable stories.

Riedl and Young's IPOCL planner (2010) narrowed the definition to include only stories where characters act intentionally. While there is still a goal for the narrative itself—what we call the author's goal—each character has its own goals, and every action a character takes must contribute to a plan to achieve one of those goals.

Ware and Young's Glaive planner (2014) broadened the definition of intentionality ($I \cap C$ in Figure 1). Instead of requiring that every action in a story contribute to a character goal, it is enough that an action *could have* contributed. In

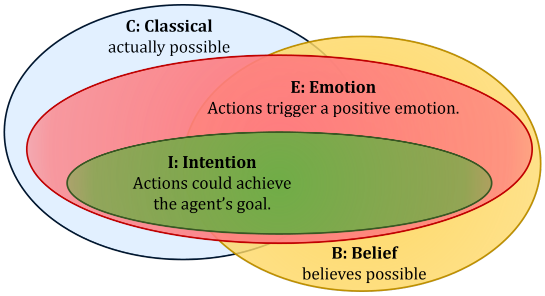

Figure 1: Character plan spaces as described by Shirvani and Ware (2020b).

other words, if there exists a version of the story where the action contributed to the characters' goals, then the action is explained, even if parts of the hypothetical story do not occur in the actual story. Teutenberg and Porteous (2013) propose a similar definition for their IMPRACTical planner.

Shirvani, Farrell, and Ware (2018) modified the space of valid character plans to reason about beliefs (B in Figure 1). A character's plan is only valid if it contributes to their goals, if they believe it is possible, and if they believe the other characters the plan depends on will act in the expected way ($I \cap B$ in Figure 1). The Sabre planner (Ware and Siler 2021) uses this model, and others have reasoned similarly about beliefs (Teutenberg and Porteous 2015; Sanghrajka, Young, and Thorne 2022).

Shirvani, Ware, and Baker (2023) further broadened the definition to state that character actions make sense any time they contribute to feeling a positive emotion ($E \cap B$ in Figure 1), which include both achieving things characters want (joy, or an increase in utility) and preventing things characters want to avoid (relief, or preventing a decrease in utility). Marsella and Gratch's EMA (2009) uses a similar definition.

In this paper, we narrow Shirvani, Ware, and Baker's definition of a valid character plan to include only *optimal* positive emotions. In other words, when characters have two available plans that are good, but one is better, we assume they will only act according to the better plan. Our approach borrows concepts from multiobjective optimization (Jahan, Edwards, and Bahraminasab 2016); treating risks and rewards as separate objectives is sometimes used in an economic context for portfolio optimization (Unni, Ongsakul, and Nimal 2020). Our approach also borrows from game tree search (Marsland 1986), which has been applied in a narrative generation context before by Kartal, Koenig, and Guy (2014) and Magnenat et al. (2022); they use Monte Carlo Tree Search to compute fast approximations of optimal character decisions, while our approach focuses on exact determination of actions known to meet our optimality definition.

3 Application Background: *Traffic Stop*

We developed this model of rational character behavior for the *Traffic Stop* virtual reality de-escalation training simulation. Police officers play the simulation, get feedback on what went well or poorly, and then replay to improve the ending. We will use the simulation's scenario as a running example throughout Section 4.

The player is a police officer who has pulled over a car for erratic driving and must either deliver a citation or let the driver off with a warning. There is a passenger in the car as well. If the player checks the driver's ID in the police database, they will notice if the driver has a protective order (sometimes called a restraining order) on file.

The experience manager can choose several elements of the state. The driver may have a protective order or not. The passenger may be the person named on the order or not. The passenger may be armed or unarmed. The player's goal is to determine whether the driver is safe, to issue a citation or warning if so, to arrest the passenger if not, and to respond to any threats that arise.

4 Character Behavior Model

Narrative planners reason on multiple levels: Plans from a character's perspective are generated to determine which actions would be rational for the character to take. The final story is a plan from an authorial perspective consisting only of character-rational actions. The model introduced in this section operates on a character level; after it has been used to determine which actions are character-rational, an authorial-level model such as the experience manager in Section 5 can select from among those actions to execute.

We will first define a series of concepts to relate plans to character preferences, and then we will introduce an algorithm for choosing the final set of available actions.

Planning Definitions

A narrative planning domain defines variables that describe the story world, such as the location and status of all objects. It also defines a set C of characters, special objects which can have beliefs and intentions.

A story graph (Riedl and Young 2005) is a directed graph representing possible plan trajectories for the narrative planning domain. A node s in a story graph is a world state, which is any function that can determine whether a Boolean logical proposition is true or false. States track the value currently assigned to each variable as well as each characters' beliefs. In the *Traffic Stop* domain, a state can answer whether the passenger is armed, whether the officer believes that the passenger is armed, whether the passenger believes that the officer believes that the passenger is armed, etc. When the world is in state s, we use $\beta(c, s)$ to denote the state character c believes the world to be in. The details of how beliefs are handled is not directly relevant to this paper, so we refer readers to Shirvani, Ware, and Farrell (2017).

A narrative planning domain defines actions that can change the world state. Actions are based on the Action Definition Language (Pednault 1994) with some additions. Every action a defines $\textsc{pre}(a)$, a logical proposition which must be true in the state before it occurs, and $\textsc{eff}(a)$, a logical proposition that must be true in the state after it occurs. For narrative planning, every action also defines a set $\textsc{con}(a)$ of characters $\in C$ who must have a reason to take the action, referred to as the action's *consenting characters*. Actions also define how character beliefs change as a result, and we omit those details here. In short, if a character observes an action their beliefs are updated, and otherwise their beliefs stay the same.

In the *Traffic Stop* domain, the officer can give a citation to the driver. The precondition of this action is that the officer and passenger are both alive and free, and the officer has the citation. The effect is that the driver now has the citation. The officer and driver are both consenting characters, meaning this action can only happen if both the officer and driver have a reason to take it.

A story graph may have an edge $s_1 \xrightarrow{a} s_2$ from node s_1 to node s_2 via action a if $\text{PRE}(a)$ is true in s_1, and s_2 is the state that would result from taking action a. We use $\alpha(a, s)$ to mean the state after taking action a in state s. So when a graph has an edge $s_1 \xrightarrow{a} s_2$, then $\alpha(a, s_1) = s_2$. If a's precondition is not true in s, $\alpha(a, s)$ is undefined.[1] We also use $\alpha(\{a_1, a_2, ..., a_n\}, s)$ to denote the state after taking the sequence of actions $\{a_1, a_2, ..., a_n\}$.

Our planning system finds a subset of the story graph consisting only of actions that are *explained* for the characters who take them, a concept that will be defined formally over the course of this section. The desired behavior is that characters take actions they believe will help them most effectively achieve world states that are better, and/or prevent ones that are worse, according to their preferences. These preferences are defined by utility functions, which map world states to real numbers. For every character $c \in C$, let $U_c(s)$ be c's utility in state s.

The utility functions in the *Traffic Stop* domain are complicated, but for our examples we will simplify them like so:

- The default utility is 0.
- A dead character has a utility of -2.
- An arrested character has a utility of -1.
- If the passenger is named on the protective order they are dangerous. A dangerous passenger has a utility of 1 when the driver is dead.
- The officer gets a utility of 1 for arresting a dangerous passenger.
- If the driver is home safely with a warning, the driver's utility is 2 and the officer's is 1.
- If the driver is home safely with a citation, the driver's utility is 1 and the officer's is 2.

The simulation ends when anyone is killed or arrested or when the driver leaves safely with the citation or warning. Note both the citation and the warning endings are positive for the driver and the officer, but the driver prefers the warning and the officer prefers the citation.

Helping and Hindering Outcomes

An action sequence $\pi = \{a_1, a_2, ..., a_n\}$ is *expected* by character c in state s just when:

1. $\alpha(\pi, \beta(c, s))$ is defined; and
2. every action a_i is explained for all $c_i \in \text{CON}(a_i)$ in state $\alpha(\{a_1, a_2, ..., a_{i-1}\}, s)$.

In other words, a character c can expect an action sequence when (1) the character believes it can occur and (2) every action in that sequence makes sense for all characters taking the action. We will now introduce more concepts that build up to a definition of "explained" as used in requirement (2).

An action a_1 *helps utility* u for character c in state s iff there exists an action sequence $\pi = \{a_1, ..., a_n\}$ such that:

1. π is expected by c in s;
2. $U_c(\alpha(\pi, \beta(c, s))) = u$; and
3. no strict subsequence of π exists that also meets these criteria.

An action helps a utility for a character if the character can anticipate a plan, containing no unnecessary actions, that starts with that action and results in that utility. The officer believes they can give the citation to the driver, they believe the driver will accept it, they expect the plan to achieve a utility of 2, and they can't imagine a simpler version of this plan which achieves the same utility.

An action a_1 *hinders utility* u for character c in state s iff there exists an action sequence $\pi = \{a_1, ..., a_n\}$ such that:

1. π is expected by c in s;
2. no action sequence π' exists such that $U_c(\alpha(\pi', \alpha(\pi, \beta(c, s)))) = u$; and
3. no strict subsequence of π exists that also meets these criteria.

An action hinders a utility if the character can anticipate a plan using that action where, after that plan, the utility is no longer reachable. If the officer arrests the passenger, they hinder utility -2 because the passenger can no longer attack them; however they also hinder utility 2 because the simulation will end without them delivering the citation.

Some existing systems use these as definitions of an explained action directly: Sabre (Ware and Siler 2021) requires actions to help a higher utility for the consenting characters, and the Shirvani and Ware (2020b) model used in the previous version of *Traffic Stop* (Fisher, Siler, and Ware 2022) also allows actions that hinder a lower utility. We introduce more specific requirements.

An action a is *better* than an action a' for character c in state s if a helps a utility higher than any that a' helps. An action a is *safer* than an action a' for character c in state s if a hinders a utility lower than any that a' hinders. An action a *dominates* an action a' for character c in state s if:

1. a is better or safer than a' for c in s; and
2. a' is neither better nor safer than a for c in s.

With respect to a set A of actions, we say an action $a \in A$ is *undominated* if no member of A dominates a.

Finally, action a is *explained* for character c in state s if:

1. a helps a utility $u > U_c(s)$ or hinders a utility $u < U_c(s)$ for c in s; and
2. no action dominates a for c in s.

[1] An exception is when an action a occurs in a state s where a character c had not believed the action was possible; i.e., a's precondition is not met in $\beta(c, s)$. In that case, rather than being undefined, $\alpha(a, \beta(c, s))$ is defined in a way that lets c's beliefs accommodate the unanticipated action. See Shirvani, Ware, and Farrell (2017) for details.

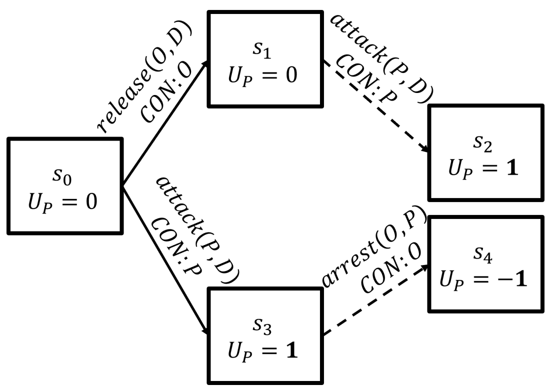

Figure 2: A story graph fragment where the passenger (P) wants to attack the driver (D), but the presence of the officer (O) affects the safety of doing so immediately. Solid arrows are edges for immediate actions from the current state of s_0 while dotted arrows are for subsequent expected actions. For conciseness, only the passenger utilities are shown.

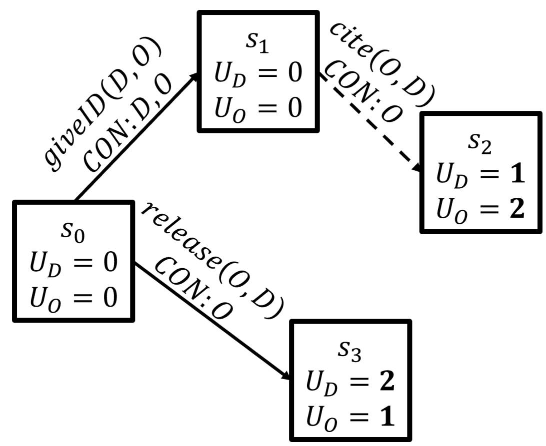

Figure 3: A story graph fragment where both the officer (O) and driver (D) want to finish the traffic stop, but have different preferences for how.

In other words, a character will take an action only if it is part of a plan to reach a state that is preferred to the current one, or to reach a state where an unwanted outcome cannot happen, and they reject one action if another action does better in one aspect without doing worse in the other.

Note that the definitions of expected sequences and explained actions are recursive; for an action to be explained for a character, the character may need to expect certain actions from other characters in order to help or hinder a utility, and for the sequence to be expected, those other characters' actions in turn need to be explained.

Suppose the passenger is armed and dangerous, but the passenger believes the officer believes they are harmless. Figure 2 shows two plans for the passenger to achieve their goal of hurting the driver: attack after the driver returns home, or attack immediately. If they attack where the officer can see, they briefly achieve a utility of 1, but would then expect to be arrested, ultimately resulting in -1. If they wait until the traffic stop is over, they can attack without repercussions. Neither plan is *better* for the passenger in terms of potential to gain utility, but the second option is *safer* because it hinders -1. Waiting for the officer to release the driver dominates attacking immediately, even though the passenger is not a consenting character to the release. This ability of characters to anticipate the consequences of their actions and the actions of others is the most significant improvement of our model over previous ones.

So far, the definition of a dominated action is circular in some cases. Suppose the passenger is harmless and there are two ways the traffic stop could be resolved, as shown in Figure 3. The officer can let the driver off with a warning (which requires only the officer's consent), or the driver can give their ID to the officer (which requires the consent of both) so the officer can print and deliver a citation. Both endings are better than the starting state for both characters, but the driver prefers to get a warning and the officer prefers to deliver the citation.

The actions that are explained in this situation depend on what the officer and driver expect the other is willing to do. Giving the ID is explained for the driver if and only if they do not expect to be let off with a warning. Giving a warning is explained for the officer if and only if the officer does not expect the driver to provide the ID.

The ambiguity leads to a dilemma. If we make a starting assumption that the driver does not expect a warning, then it is explained for the driver to give their ID. If we make a starting assumption that the officer does not expect the driver to give the ID, then it is explained for the officer to issue a warning. However, if we make a starting assumption that the driver expects the officer to issue a warning and that the officer expects the driver to give the ID, then the characters become deadlocked and the narrative cannot progress. We will next extend our model with an algorithm for determining a node's set of explained actions in a way that avoids deadlocks.

Pareto-Based Action Pruning

We determine which actions are explained by iteratively detecting and pruning (i.e., eliminating) unexplained actions. Let s be some node in the story graph that we want to prune. Initially, s has all possible out-edges. That is, there is an edge $s \xrightarrow{a} s'$ for every action a whose precondition is met in s. We assume that all descendant states of s have already undergone the pruning process. In other words, any plan that a character considers starting in state s will be explained, except for possibly the first step.

The pruning process works by allowing characters to approve actions they want to take. Once an character approves a set of actions in one step they will not approve any other actions in later steps. If an action gets no approvals, it is pruned.

The process below prunes one action at a time. Each time an action is pruned, the process starts over at the beginning from Step 1. This is because the range of possible outcomes changes each time an action is pruned, so which actions are dominated or undominated may also change.

1. Every action must be better or safer (or both) for all of its consenting characters. If an action can be found which is neither better nor safer for one of its consenting characters, prune that action and start over at Step 1.
2. For every character c we find the set $Pareto(c)$ of actions that are undominated for c. We consider all possible unpruned actions, regardless of who consents.
3. A *unanimous action* is one that is in $Pareto(c)$ for all of its consenting characters. Unanimous actions are the clear best choices. Every character for whom a unanimous action is in $Pareto(c)$ approves those unanimous actions.[2]
4. A *compromise action* is one for which (1) none of its consenting characters have approved other actions; (2) the action is better or safer for all consenting characters; and (3) the action is in $Pareto(c)$ for at least one consenting character. Every character who is a consenting character to a compromise action or for whom a compromise action is in $Pareto(c)$ approves those compromise actions.[2]
5. At this point, if a character has not approved any actions then their best outcomes depend on other characters doing suboptimal things, so they should reconsider their options. We now recalculate $Pareto(c)$ for all characters, but this time we consider only actions where c is a consenting character, instead of all actions like in Step 2. Note that these new sets may change which actions are unanimous and compromise actions.
6. Every character for whom a unanimous action is now in $Pareto(c)$ approves those unanimous actions.
7. Every character who is a consenting character to a compromise action or for whom a compromise action is now in $Pareto(c)$ approves those compromise actions.
8. Find one action that has received no approvals. Prune it, and start over at Step 1. If no actions can be found to remove, pruning is complete.

It is possible a character has no actions remaining after pruning. It is also possible a character has many actions remaining. This is deliberate. Our goal is not to decide on exactly which actions characters should take, but only to remove actions that would not appear believable. Once non-believable actions have been removed, it is up to the experience manager of the narrative to decide what should happen based on its goals for the story.

Returning to the example in Figure 3, initially $Pareto(Officer)$ consists only of the driver giving their ID, because it is better for the officer; and conversely, $Pareto(Driver)$ consists only of being released with a warning. The algorithm determines that no actions are unanimous because each dominates the other for one of its consenting characters. However, the give-ID action is undominated for one consenting character, the officer, and still leads to a utility increase for the other consenting character, the driver, so it is marked as a compromise action and retained while the release action is pruned.

If we imagine a version of the domain where giving the ID has only the driver as a consenting character, then giving the ID is no longer a compromise action. Because $Pareto(Driver)$ has no unanimous or compromise actions, the algorithm recomputes $Pareto(Driver)$ from only the set of actions where the driver is a consenting character. The release action is dropped from the driver's consideration, making the give-ID action undominated, so giving becomes the new sole member of $Pareto(Driver)$ and is marked as unanimous. A similar recalculation of $Pareto(Officer)$ leads to the release action being marked as unanimous as well. Both actions are retained.

Practical Considerations

Our definition of action dominance requires finding a plan to show an action is better or proving no plan exists to show an action is safer. In practice, when pruning a story graph, we place two upper limits on this process. The first limits how long a plan can be. The second limits how far past a plan we can search to check whether an outcome has become impossible. These limits can be tuned based on the problem to find a balance between the rationality of characters and the speed of pruning.

5 *Traffic Stop* Integration

The *Traffic Stop* police de-escalation training simulation, illustrated in Figure 4, is the first deployment of our model in practice. It features a scenario similar to the planning domain described in previous examples, with the driver and passenger as NPCs. The player plays the role of a police officer who has pulled over a car for erratic driving and whose objective is nominally to issue a citation to the driver. Complications can arise such as the characters' non-cooperation or the existence of a protective order by the driver against an unidentified party, who may or may not be the passenger.

The *Traffic Stop* software architecture is divided into two subsystems: the environment and the experience manager. The environment includes the components that display the game world to the player, detect player interactions with objects in the world, and manage low-level decompositions of high-level actions using the Camelot (Shirvani and Ware 2020a) action framework. The experience manager is responsible for the system's intelligent decision-making—namely, choosing the high-level actions for the NPCs—to ensure that the system's pedagogical goals are met. The pairing of these two components is modular; our modification to the system consisted of implementing a new experience manager using our planning model.

Our design objectives centered around *Traffic Stop*'s original purpose to help trainees practice resolving high-pressure

[2]Note unanimous and compromise actions can be in the Pareto set for characters other than the consenting characters. For example, an armed and dangerous passenger prefers to wait for the traffic stop to end before they attack the driver, even through the passenger is not a consenting character for any of the actions between the officer and driver. In other words, the passenger will not act because their best outcome is achieved by waiting for the officer and driver to do things which eventually enable the optimal outcome for the passenger.

Figure 4: A screenshot from the player's point of view during a playthrough of *Traffic Stop*.

situations without unnecessary use of force: First, NPCs should act believably. One of the inciting incidents for developing our Pareto-dominance-based model was that without it, unduly aggressive character plans were produced. For instance, if the passenger foresaw any way that their utility could be lowered by the player, a plan would be produced for them to attack the player at the beginning of the scenario. Overprediction of danger has been cited as a major factor in excessive police use of force (Marenin 2016); a de-escalation training platform should not reinforce this pattern.

Second, players should be able to have an informative experience in only a few playthroughs. Although some de-escalation skills take extensive practice and our simulation only complements rather than replaces training programs for those skills, our focus is on enabling information-dense short-term interventions. Our experience manager ensures that players who follow best practices get good outcomes, and players who violate best practices get bad outcomes. Players can explore different decisions and receive a clear illustration of the possible consequences of those decisions.

Experience Management

We used an experience management paradigm introduced by Ware et al. (2022): The experience manager models the playthrough as navigation of a precompiled story graph. The story graph is constructed first by exploring the whole state space for a planning problem to get a graph of all classically legal plans, and then applying rules to prune the graph. One of the first pruning steps is to remove NPC actions that are not explained according to our model. This addresses our design objective of character believability, as the remaining actions are more aligned to the NPCs' goals. Further pruning steps choose between those actions in a way that is most convenient to the design objective of matching player outcomes to player choices.

The final story graph represents a policy where there is at most one surviving NPC action for each state, so the experience manager's decisions at runtime are predetermined:[3] In states where all NPC actions have been pruned, the experience manager waits for the player to act. In states with an unpruned NPC action, the experience manager tries to execute that action. However, because the player and NPCs are interacting in real time, the experience manager's attempted action may be interrupted by the player's.

To prepare the final story graph, we first generated unpruned story graphs for several versions of a planning problem with differences in the initial state: The passenger could have a gun or be unarmed; the driver could have an active protective order against someone or not, which can be revealed when the driver's ID is looked up in the police database; and the name on the protective order could be the passenger's name or a different name.

We pruned each story graph to remove unexplained NPC actions using the definition of explanation presented in this paper. We then combined the story graphs into one *superposed* story graph (Robertson and Young 2018) where a story graph node can represent multiple possible worlds at a time. For instance, at the beginning of the scenario, the experience manager leaves the existence and target of the protective order undecided. If the player starts by acquiring and looking up the driver's ID, at that moment the experience manager determines whether the protective order exists and chooses an arbitrary name to appear on the protective order. If the player later acquires the passenger's ID, at that moment the experience manager chooses the passenger's name as either the one on the protective order or a different name.

This approach gives the experience manager an additional tool to ensure the player experiences an outcome that highlights any mistakes. For instance, suppose the player tries to investigate whether the passenger is the person on the protective order, but the passenger refuses to provide their ID. The best practice in this situation, as described to us by a consulting officer, is to explain to the passenger why the ID is being requested. If the player instead assumes the passenger is a criminal and arrests them, the experience manager can retroactively decide that the passenger was innocent, so the player's mistake will never be rewarded by luck.

To ensure appropriate endings, we identified a mapping from situations to acceptable player actions, and pruned the combined story graph to remove any paths where the player would get a good ending after taking an unacceptable action, or a bad ending after taking only acceptable actions. Finally, to reach a policy of at most one NPC-only action per state, we arbitrarily pruned NPC-only actions from states where multiple remained.

We are investigating ways to do effective quality assurance on large story graphs. In this case, our process combines quality assurance with the construction of the story graph itself by detecting and removing undesirable paths.

Study Procedure

We evaluated our experience manager in a human-subjects study with volunteers from the University of Kentucky Po-

[3] Currently, a human operator passes commands between the experience manager and environment. This is needed because certain actions involve verbal communication by the player and *Traffic Stop* does not yet have a speech processing component that can accurately map utterances to actions. The operator is trained to interpret the player's utterances and otherwise to follow the experience manager policy deterministically, simulating a fully automated experience manager with an idealized speech processor.

lice Department.[4] Six officers participated, each with between 8 and 35 years of policing experience. All officers had prior traffic enforcement experience, and none had previously used the *Traffic Stop* simulation.

Each officer began their session by viewing a video explaining the purpose of the simulation, instructions on using the virtual reality equipment and game controls, and safety guidelines. The officer then played a tutorial to familiarize them with game mechanics, followed by playing the main scenario repeatedly, as many times as they desired. After each playthrough, they were presented through the game interface with feedback about what they did well or poorly.

Along with gameplay logs, we collected data in the form of a survey given to participants after their last playthrough. The survey included the Presence Questionnaire, introduced by Witmer and Singer (1998) to measure a user's sense of immersion in a virtual environment. The survey also included the custom set of questions in Figure 5, modified from Ware et al. (2022), that measured other aspects of the participant's experience: how believable they found the NPCs, how much agency they felt they had, and how valuable they felt the simulation was as a training tool. Finally, the survey invited the participant to give optional freeform comments.

Character Believability

- The virtual characters felt realistic.
- The virtual characters reacted to things they saw and ignored things they did not see.
- The virtual characters tried to accomplish their goals.
- The things the virtual characters did made no sense to me. **(N)**

Agency

- My actions had a significant effect on the story.
- I understood the consequences of the choices I made.
- The story was the same no matter what choices I made. **(N)**

Training Effectiveness

- This simulation helped me feel more prepared for a situation I might experience.
- Exercises like this are a bad way to train police officers. **(N)**
- I want more training exercises like this.

Figure 5: The Simulation Experience Questionnaire. Participants were asked to rate each statement (bulleted) on a five-point Likert scale. The items marked **(N)** are reverse-coded. The category headers were not shown as part of the questionnaire and are included to show which questions were grouped together to compute subscores.

[4] Our study procedure was approved by our institution's IRB.

Survey Results

The Presence Questionnaire is scored by taking the average of the participant's Likert-scale responses from 1 to 7 for each question; it also has subcategories whose scores can be taken in a similar manner. Figure 6 shows the average of all participants' overall and subcategory scores, as well as the scores for each individual participant. Note that each bar represents the mean of responses to several questions. Participants tended toward agreement with the presence questions, with only one officer having an overall score and some subcategory scores below the scale midpoint of 4.

We defined subcategories similarly for our simulation experience questionnaire: one for questions about believability of character behavior, one for questions about the participant's agency to affect the story, and one for questions about the value of the simulation as a training exercise. Figure 7 shows the scores for this questionnaire. The average across all participants was above the scale midpoint of 3 for both the overall and subcategory scores. Individual scores were above the scale midpoint for a majority of participants, except within the agency subcategory, suggesting that an area for further improvement is to have the simulation more clearly illustrate the relationship between the officer's actions and the story outcome.

The participants' freeform comments in the survey emphasized ways to improve the interface (e.g., "use a glove instead of hand controllers") or the realism of the scenario (e.g., "include descriptors of suspects").

Gameplay Results

There were endings, ranked from worst to best, that a participant could achieve; a police chief NPC gave praise in good endings, while bad endings included the player getting injured, or the chief pointing out harms to civilians. For each participant, Figure 8 shows the endings they achieved for each playthrough chronologically. All participants achieved the best ending at least once: One officer achieved the best ending in every playthrough, another achieved the best ending initially and later had a suboptimal playthrough (which may be the result of exploring the simulation after already mastering the ideal path), and the rest started with suboptimal playthroughs and later achieved the best ending. The Kendall rank correlation between playthrough number and ending ranking is positive (0.42, $p = 0.03$).

This study provides preliminary evidence that our application helps users become more skilled with repeated play, and is received positively by those users.

6 Limitations

The main limitation of the planning model itself is its computational intensity. Even with limits on the length of expected action sequences a character considers, it would be impractical to call a planner using our model at runtime during gameplay like more lightweight narrative planners (Porteous, Cavazza, and Charles 2010; Ware and Young 2015), instead being restricted to offline generation. See Ware et al. (2022) for a discussion of the scalability of offline methods.

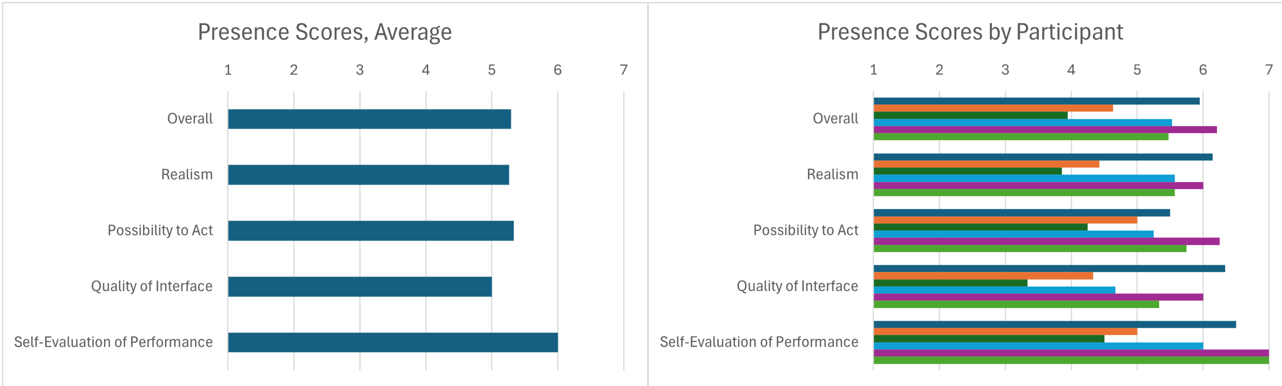

Figure 6: Presence questionnaire score and subcategory scores.

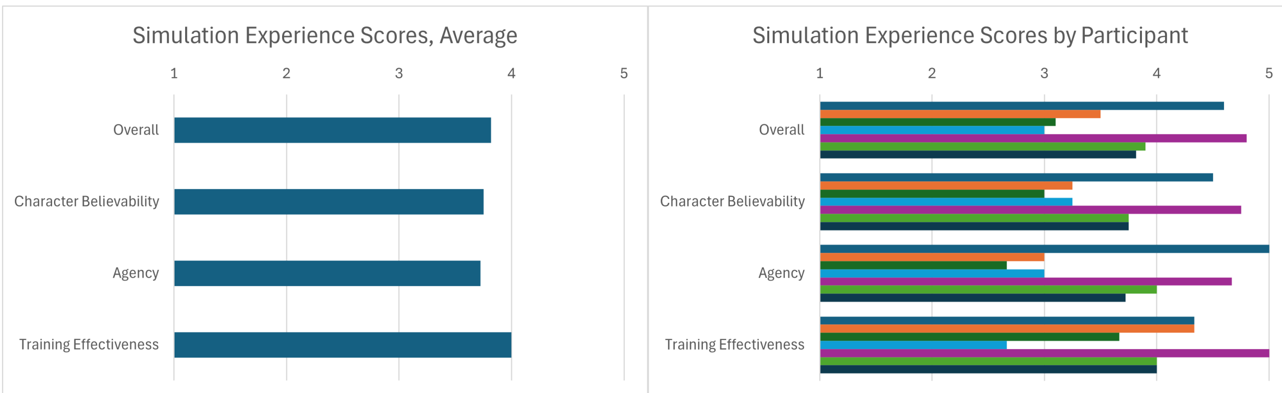

Figure 7: Simulation experience questionnaire score and subcategory scores.

The limitations of our evaluation include its lack of controls and its narrow participant pool. Although the survey results look positive subjectively, further experiments would be needed to establish whether the new *Traffic Stop* experience manager improves over other approaches, and whether our planning model is a cause of those improvements. These experiments should also include inexperienced officers and should measure the longer-term impacts of the training. Finally, we have not yet evaluated our character model in a more general context than the *Traffic Stop* application.

7 Conclusions

This paper introduced a narrative planning model where characters reason like rational game-playing agents, pursuing the best outcomes and avoiding the worst outcomes they can, based on the actions they anticipate from other characters. It is efficient enough for offline compilation of story graphs to later be used by a game, while avoiding some of the problematic character behaviors of simpler models. For interactive narratives, the offline approach to character reasoning is complementary to offline approaches to narrative mediation (Riedl and Young 2005), the process of ensuring that unexpected player decisions will still lead to an experience matching the designer's intent.

We showed how our model was used to develop a serious-games application that, based on a small user study, is effective at teaching users how to interact with the virtual characters. Besides this anecdotal support, we are planning a larger, controlled study of the model separate from the application, similar to Shirvani, Farrell, and Ware (2018), that compares stories generated by our model to stories generated by previous models in terms of character believability.

A Pareto-based model of character planning does not need to be restricted to increasing or preventing decreases to one utility function. Consider that a character's utility function may embed multiple objectives—e.g., in the traffic stop planning domain, the officer character gets utility based on their own survival, their adherence to the law, other characters' survival, and the completion of the traffic stop. By splitting these into multiple utility functions per character, we could apply the same planning process for selecting actions from the Pareto front of the different utility functions. This would allow us to simulate internal conflicts between desires, personality traits, or social norms (Schroeder 1999).

Acknowledgments

We thank Kishonna Gray, Jennifer Melgar, and the University of Kentucky Police Department for consultations on the traffic stop simulation.

This material is based upon work supported by the U.S.

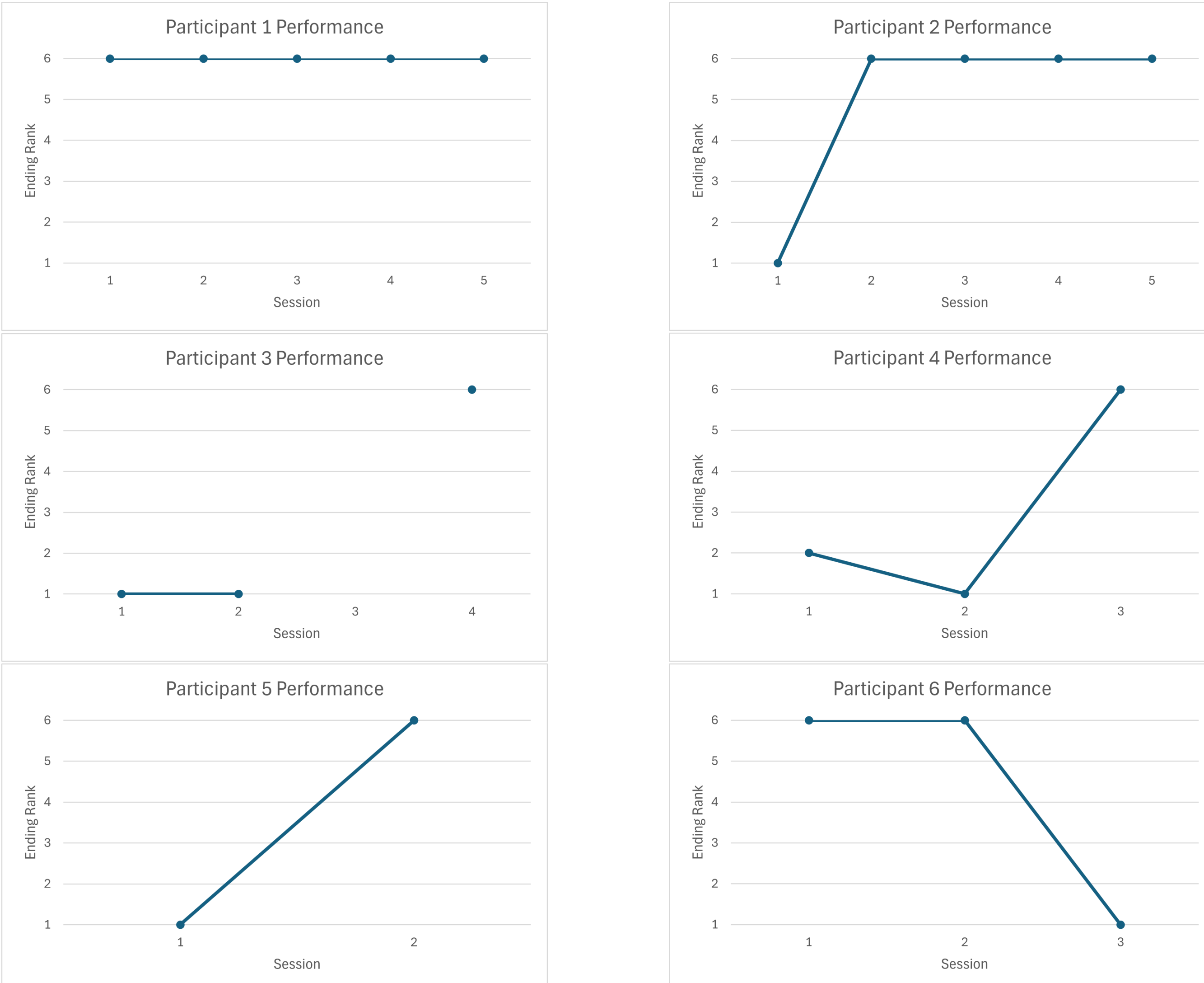

Figure 8: The ending ranks achieved by subjects over repeated playthroughs, where 6 is the best and 1 is the worst possible ending. A data point is missing for one participant due to a bug preventing the session from being finished.

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

References

Alhussain, A. I.; and Azmi, A. M. 2021. Automatic story generation: A survey of approaches. *ACM Computing Surveys*, 54(5).

Fikes, R. E.; and Nilsson, N. J. 1972. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3): 189–208.

Fisher, M.; Siler, M.; and Ware, S. G. 2022. Intelligent de-escalation training via emotion-inspired narrative planning. In *Intelligent Narrative Technologies Workshop*, volume 13.

Jahan, A.; Edwards, K. L.; and Bahraminasab, M. 2016. *Multi-Criteria Decision Analysis for Supporting the Selection of Engineering Materials in Product Design*. Butterworth-Heinemann.

Kartal, B.; Koenig, J.; and Guy, S. J. 2014. User-driven narrative variation in large story domains using Monte Carlo tree search. In *International Conference on Autonomous Agents and Multiagent Systems*, volume 13, 69–76.

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

Magnenat, S.; Raymond, H.; Enderlin, D.; Knobloch, S.; and Sumner, R. W. 2022. NPC Engine: A high-performance, modular, and domain-agnostic MCTS framework for emergent narrative. In *Intelligent Narrative Technologies Workshop*, volume 13.

Marenin, O. 2016. Cheapening death: Danger, police street culture, and the use of deadly force. *Police Quarterly*, 19(4): 461–487.

Marsella, S. C.; and Gratch, J. 2009. EMA: A process model of appraisal dynamics. *Cognitive Systems Research*, 10(1): 70–90.

Marsland, T. A. 1986. A review of game-tree pruning. *ICGA Journal*, 9(1): 3–19.

Meehan, J. R. 1975. Using planning structures to generate stories. *American Journal of Computational Linguistics*, 2(4): 78–94.

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

Porteous, J.; Cavazza, M.; and Charles, F. 2010. Applying planning to interactive storytelling: Narrative control using state constraints. *ACM Transactions on Intelligent Systems and Technology*, 1(2): 1–21.

Riedl, M. O.; and Young, R. M. 2005. From linear story generation to branching story graphs. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 111–116.

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

Robertson, J.; and Young, R. M. 2018. Perceptual experience management. *IEEE Transactions on Games*, 11(1): 15–24.

Sanghrajka, R.; Young, R. M.; and Thorne, B. 2022. HeadSpace: Incorporating action failure and character beliefs into narrative planning. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, 171–178.

Schroeder, M. 1999. How to tell a logical story. In *Narrative Intelligence: Papers from the AAAI Fall Symposium*. AAAI Press.

Shirvani, A.; Farrell, R.; and Ware, S. G. 2018. Combining intentionality and belief: Revisiting believable character plans. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 222–228.

Shirvani, A.; Ware, S.; and Farrell, R. 2017. A possible worlds model of belief for state-space narrative planning. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 13, 101–107.

Shirvani, A.; and Ware, S. G. 2020a. Camelot: A modular customizable sandbox for visualizing interactive narratives. In *Intelligent Narrative Technologies Workshop*, volume 12.

Shirvani, A.; and Ware, S. G. 2020b. A formalization of emotional planning for strong-story systems. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 116–122.

Shirvani, A.; Ware, S. G.; and Baker, L. J. 2023. Personality and emotion in strong-story narrative planning. *IEEE Transactions on Games*, 15(4): 669–682.

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

Teutenberg, J.; and Porteous, J. 2015. Incorporating global and local knowledge in intentional narrative planning. In *International Conference on Autonomous Agents and Multiagent Systems*, 1539–1546.

Unni, A. C.; Ongsakul, W.; and Nimal, M. M. 2020. Fuzzy-based novel risk and reward definition applied for optimal generation-mix estimation. *Renewable Energy*, 148: 665–673.

Ware, S. G.; Garcia, E. T.; Fisher, M.; Shirvani, A.; and Farrell, R. 2022. Multi-agent narrative experience management as story graph pruning. *IEEE Transactions on Games*, 15(3): 378–387.

Ware, S. G.; and Siler, M. 2021. Sabre: A narrative planner supporting intention and deep theory of mind. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 99–106.

Ware, S. G.; and Young, R. M. 2014. Glaive: A state-space narrative planner supporting intentionality and conflict. In *AAAI 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 AI in Games*, 8(4): 402–411.

Witmer, B. G.; and Singer, M. J. 1998. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3): 225–240.

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