Another Look at Search-Based Drama Management

(Short Paper)

Mark J. Nelson
College of Computing
Georgia Institute of Technology
mnelson@cc.gatech.edu

Michael Mateas
Computer Science Department
University of California, Santa Cruz
michaelm@soe.ucsc.edu

ABSTRACT

A drama manager (DM) is a system that monitors an interactive experience, such as a computer game, and intervenes to keep the global experience in line with the author's goals without decreasing a player's interactive agency. In declarative optimization-based drama management (DODM), an author declaratively specifies desired properties of the experience; the DM intervenes in a way that optimizes the specified metric. The initial DODM approach used online search to optimize an experience-quality function. Later work questioned both online search as a technical approach and the experience-quality optimization framework. Recent work on targeted trajectory distribution Markov decision processes (TTD-MDPs) replaced the experience-quality metric with a metric and associated algorithm based on targeting experience distributions. We show that, though apparently quite different on the surface, the original optimization formulation and TTD-MDPs are actually variants of the same underlying search algorithm, and that offline cached search, as is done by the TTD-MDP algorithm, allows the original search-based systems to achieve similar results to TTD-MDPs. Furthermore, we argue that the original idea of optimizing an experience-quality function does not destroy interactive agency, as had previously been argued, and that in fact it can capture that goal directly.

1. INTRODUCTION

Interactive drama is an interactive experience in which a player interacts with a rich story world in a way that gives a feeling of strong interactive agency while creating, as a result of those interactions, a narrative experience that is dramatic, interesting, and coherent. Putting the player in a story world populated by believable agents does not necessarily create interactive drama: An interactive drama must be designed such that the series of agent-player interactions results in a globally coherent and interesting narrative. A drama manager (DM) is a central coordinator that directs and adapts the agents and other contents of a story world as an experience unfolds to maintain global narrative goals, without removing the player's interactive agency.

One approach is declarative optimization-based drama management (DODM). In DODM, the author specifies a list of the narratively important events that could occur in the experience, called *plot points*; a set of *DM actions* that the DM can take to intervene in the experience; and an *evaluation function* that rates the quality of complete experiences.

Plot points include things such as the player engaging in a particular conversation with an agent in the story world or acquiring an object. They hvae ordering constraints that capture the physical possibilities of the story world. For example, a player cannot interact with a genie in a lamp without having first found the lamp. Plot points are also annotated with information that may be useful to the evaluation function, such as where it happens. DM actions can *cause* a plot point to happen, *hint* to make it more likely that it will happen, *deny* it so it cannot happen, or *undeny* a previously denied plot point. For example, the DM might tell a non-player character to go up to the player and reveal some information, causing the plot point in which the player gains the information. The set of plot points and DM actions, when combined with a player model, provides an abstract, high-level view of the unfolding experience. The evaluation function takes this view of a completed experience and assigns it a rating. The drama manager's job is then to optimize its use of DM actions so as to maximize this evaluation.

The original DODM system, proposed as search-based drama management (SBDM), used a search algorithm to maximize this experience evaluation function [1, 9]. Recent work has questioned both the technical feasibility of search as the optimization method [5], and the conceptual usefulness of having a DM maximize an experience-quality function [8]. In particular, Targeted Trajectory Distribution Markov Decision Processes (TTD-MDPs) have proposed a new goal, with associated algorithms, of targeting an author-specified distribution of experiences [8, 2].

We revisit these criticisms. We show that, although they appear quite different as originally described, the SBDM and TTD-MDP algorithms are actually variants of the same underlying search algorithm. Furthermore, when the original search algorithm is enhanced by caching, as the TTD-MDP one is, it performs at the same level. As a conceptual matter, we argue that the original idea of optimizing an experience-quality function rather than targeting an experience distribution does not destroy player agency, and that to the contrary an experience-evaluation function can directly include interactive agency as a goal, whereas simply adding nondeterminism via TTD-MDPs does not.

Cite as: Another Look at Search-Based Drama Management (Short Paper), Mark J. Nelson and Michael Mateas, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1293-1296.
Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

```
Build a large tree of possible experience trajectories
for all nodes n in a post-order (leaf-first) traversal do
  if n is terminal then
    n.value ← terminalValue(n)
  else
    n.policy ← optimize(n.actions, n.children)
    n.value ← backup(n.children, n.policy)
  end if
end for
```

Figure 1: Pseudocode for generic cached search. The TTD-MDP and SBDM algorithms share this structure, but differ in how they define terminal values, carry out the optimization, and perform backups.

2. SBDM AND TTD-MDP

DODM was proposed and developed by Bates [1] and Weyhrauch [9] as search-based drama management (SBDM). They proposed a game-tree-search analogy: the player makes "user moves" (*plot points*) through their interaction with the game world, and the DM responds with its own "system moves" (*DM actions*). The DM chooses its "moves" using an author-supplied experience quality function that rates completed experiences, and expectimax search. The expectimax search alternates between maximizing over the available DM actions, and averaging over the possible plot points that could follow, weighted according to a model of likely player behavior. A fairly simple player model is used: the player is assumed to be equally likely to make each of the next possible plot points happen, except for those which have been hinted at, which are considered more likely by a multiplier that the author specifies in an annotation to the hint. To keep things tractable, a sampling search, called SAS+, is used past a certain depth.

Roberts *et al.* [8] proposed a change to the basic formulation. They argued that when the goal is to maximize an evaluation function, the only source of gameplay variation will be unpredictability on the part of the player—and that given sufficiently powerful DM actions, the DM could force an "optimal" story on the player, destroying the truly interactive aspects of the experience. They therefore proposed to start with a desired distribution of experiences (trajectories through the story space), and aim to use the DM actions in a way that would make the actual distribution come as close to that target as possible. Algorithmically, the TTD-MDP system builds a large tree sampled from the space of all possible trajectories; each node in the tree then solves an optimization problem to find a distribution over its available actions that will, according to the player model, cause a resulting distribution over successor plot points that is as close as possible to the distribution specified by the author.

3. OPTIMIZATION BY CACHED SEARCH

SBDM uses an online expectimax search that, to remain computationally tractable, past a cutoff search depth limit switches to sampling trajectories and averaging their evaluations instead of performing full search. The TTD-MDP algorithm [8] operates offline, sampling many possible trajectories through the story world and building them into a tree, and then solving an optimization problem at each node. When a trajectory is seen that wasn't among those sampled in the tree, it falls back to online search. These two algorithms are quite similar when SBDM also uses a tree of cached results. Both build a cached tree, perform an optimization at each node starting from the leaves and working upwards, and back results up the tree, as shown in the generic pseudocode in Figure 1. The main differences are that they choose actions at each node using a different objective function, and assign and back up values to nodes based on different evaluation criteria.

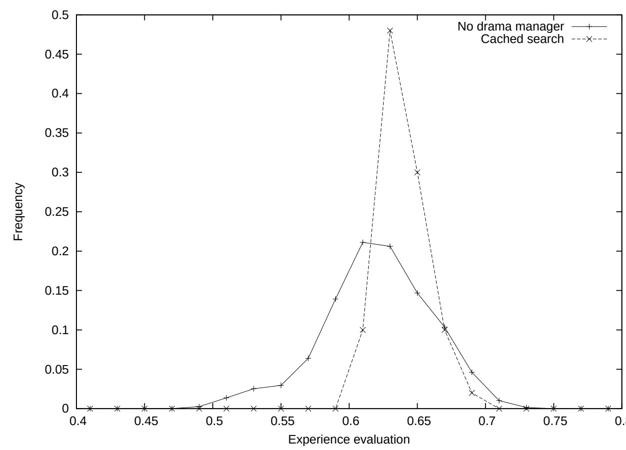

Figure 2: Frequency with which experiences of different qualities (as measured by the evaluation function) occur for a simulated user with a DM guided by iterative-deepening search assuming a minute between plot points, versus a baseline of no DM.

In SBDM, the terminal nodes have their values given by the experience-evaluation function. The policy at each node is to take the DM action that maximizes expected evaluation value when averaged over its children nodes according to the player model. The node's own value is then set to the expected value of this action. In the TTD-MDP algorithm, the terminal nodes have target probabilities as their values. The policy at each node is the distribution over actions that minimizes expected divergence from the target distribution specified by the node's children, with the expectation computed according to the player model. The node's own value is then set to the sum of its children's target probabilities.

Both algorithms can be made to adaptively fill their cached trees during gameplay, using background processor cycles between the occurence of plot points [2]. In fact, once we note the connection with search, we can consider well-known space versus time tradeoffs to avoid literally maintaining a large cache in memory at all. The tree in memory in which we fill in nodes at the frontiers is essentially breadth-first search, which has nice execution-time properties but exponentially large memory requirements. A common alternative is iterative deepening search, which performs a series of fixed-depth depth-first searches with increasing depths, stopping and returning the result of the deepest completed search when the next decision is needed.

Figure 2 shows histograms, both with and without a DM, of the frequency with which experiences of varying quality appear over a number of runs with a simulated player (the same acting-randomly-except-for-hints simulated player used by all previous work), as measured by the authorspecificied evaluation function. The DM in this setup uses a "synthetic" set of DM actions consisting of a causer, denier, and reenabler for every possible plot point; this was the hypothetical maximally powerful setup in which Nelson & Mateas [5] found that search still could not work well. One curve shows the results without a DM, and the other with the iterative-deepening DM. As can be seen by the fact that the curves move towards the right—indicating more frequent highly-rated stories and less frequent low-rated stories—cached search, as a technical matter, functions well in this story, contrasting with the previous results.

4. WHAT TO OPTIMIZE

Since the two algorithms operate similarly, the main question in deciding between SBDM and TTD-MDPs is sorting out what it is a DM should be optimizing: what constitutes a good interactive drama? The main goal is a narratively interesting and coherent experience with strong player agency.

4.1 Maximizing experience quality

DODM envisions an evaluation function that, given a completed experience (a sequence of plot points and DM actions), will rate it based on various features that the author thinks the experience should have had. This function rates the quality of *interactive experiences*, not the quality of plot-point sequences considered as if they were non-interactive stories. That is, DODM does not create interactive drama by taking a set of desiderata for *non-interactive* drama and trying to bring it about in the face of interactivity. Rather, it takes a set of desiderata for the interactive dramatic experience itself, and tries to maintain those. Some DM systems do frame the drama-management problem as one of mediating between authorial narrative goals and player freedom [10, 4, 6]. In DODM, however, the DM starts with a more general notion of what constitutes a narratively interesting experience, and intervenes when necessary to make sure the player has one.

Looking in particular at Weyhrauch's evaluation function, it specifies a number of weighted features that capture his notion of a good experience in his *Tea for Three* story world.

One group of features mainly encourages narrative coherence: *thought flow* prefers stories where subsequent actions relate to each other; *activity flow* prefers stories that have some spatial locality of action; and *momentum* prefers certain pairs of plot points that build on each other well. Separately, the *motivation* feature prefers stories in which at least some plot points are motivated by previous plot points. Note that these are preferences for the interactive experience, and would not necessarily be the same if evaluating a linear story. Weyhrauch doesn't argue that it's necessarily bad for narratives to have the action move around frequently between different locations; rather, he argues that if each plot point happens in a different location from the last in an interactive experience, it was probably the case that the experience contained a lot of uninteresting wandering around the world.

Given only these features, however, there is a danger that the system could identify certain plot-point progressions as ideal and force the player into them, defeating the goal of interactive agency. To avoid this outcome, two versions of an additional evaluation feature, one proposed by Weyhrauch and one by Nelson & Mateas, aim at encoding interactive agency, though from different perspectives.

Weyhrauch's *options* feature identifies twelve meaningful goals a player could have at various points in *Tea for Three*. For example, the goal "talk to George about the new will" is considered to be active between the time the player finds a note mentioning a new will and the time that the player either talks to George about it or is prevented from doing so by other events. The number of goals active at any given time is a rough measure of the degree of interactive agency available. The *options* feature encodes a preference for many such meaningful options to be available towards the beginning of the game, decreasing to fewer towards the end.

Nelson & Mateas's *choices* feature captures a more local notion of agency, measuring how many plot points could have followed any given point in the story, given the ordering constraints in the world and the effects of causers and deniers. If at some point only one plot point could possibly come next, then the same bit of story would play out next regardless of what the player did. If on the other hand many plot points could come next, then the player could locally influence the story to a much greater extent. The *choices* feature has the advantage that it can be computed automatically for any story, but the *options* feature has the advantage that it captures a higher-level notion of meaningful interactive agency.

Finally, a *manipulativity* feature penalizes uses of DM actions that are likely to be particularly noticeable, like clumsy hints or moving objects that the player can see. This is a meta-feature of sorts encoding a preference for the DM's operation to be invisible. Although we use agents in service of a narrative rather than merely simulating them as believable agents in their own right, we do still want them to avoid doing things that would break believability.

4.2 Targeting an experience distribution

Roberts *et al.* [8] criticize the idea of maximizing a story-quality function, arguing that an effective DM can simply bring about the same highly-rated story each time, destroying interactive agency and replayability. They propose instead that the goal of the DM should be to target a distribution of experiences, specified either by some mapping from an evaluation function (*e.g.* bad experiences should never happen, and good ones should happen in proportion to their quality), or by having the author specify a few prototype experiences and then targeting a distribution over experiences similar but not identical to the prototypes [7].

Since the goal of DODM is to maximize experience quality rather than story quality, though, an evaluation function should measure not only the quality of the story that a series of interactions produces, but also the quality of the interaction itself, including elements such as interactive agency; hence the *options* and *choices* features. Moreover, targeting a distribution of experiences does not necessarily coerce the player less than even targeting a single maximum-quality story would. With enough causers and deniers, an TTD-MDP system can directly cause its desired distribution of experiences to come about, by randomly selecting (according to the desired distribution) which DM actions to take in each play-through. Although that would vary *which* story the player is forced into each time, it still uses the DM actions to produce a specific story with no input from the player—randomly selecting a different story to force the user into each time does not create interactive agency.

Indeed we find similar levels of coerciveness if we look at the DM actions performed by the TTD-MDP based system and the SBDM system on the version of *Anchorhead* with a "synthetic" set of DM actions that Roberts *et al.* use as a point of comparison. The "synthetic" set of actions consists of a causer, denier, and reenabler for every possible plot point in the story, thus creating a hypothetical situation where the DM has a maximally powerful set of actions available. The TTD-MDP system claimed better replayability in this case, since it produced a wider variety of stories. However, both the TTD-MDP system and the SBDM system acted almost maximally coercively: they each performed an average of around 15 DM actions per experience, in an experience only 16 plot-points long. The TTD-MDP system varied which specific coercion it performed from run to run, but that again does not constitute interactive agency, which requires that the player, rather than system nondeterminism, be able to meaningfully influence the outcome.

That both systems are quite coercive, however, does point to a failure in the particular experience-quality evaluation function that both used. We can correct this by simply putting a greater weight on the *choices* feature, which emphasizes that giving the player many choices in what to do really is an important part of an interactive experience. When we increase *choices* from being 15% of the total evaluation weight to 50%, both systems drop to using an average of around 5 DM actions per experience.

How to best write evaluation functions does remain an issue that would benefit from additional experimentation in the context of specific real interactive dramas. It is worth noting that all the recent systems have focused on the "synthetic" model of *Anchorhead* that has only causers, deniers, and reenablers, and lacks the hint actions that a DM could use to provide more narrative guidance to the player without unduly removing interactive agency; by contrast, a real application would likely use hints frequently.

Whether the TTD-MDP formulation does still improve matters in a different way depends on the particular way in which the target distribution is defined, and on what we consider to be the goals of interactive drama. In the case where the target distribution is generated by a mapping from an experience-quality function, the results will be fairly similar to the results from an evaluation-function-maximizing approach, since both systems will be trying to avoid low-rated experiences and increase the probability of highly-rated ones according to the same function. The TTD-MDP approach will add some more nondeterminism in doing so; how much depends on how the mapping is constructed. Alternate ways of specifying a target distribution of experiences for TTD-MDPs, however, such as specifying several prototype experiences and inducing a distribution over experiences similar to those prototypes [7], suffer from a greater loss of interactive agency. If the player is being forced into one of several prototype experiences or minor variants thereof, the fact that the specific experience they're forced into is chosen nondeterministically does not preserve interactive agency.

4.3 Non-dramatic interactive experiences

We focus on authoring interactive drama. Similar techniques can be used for other kinds of interactive experiences, which may have different considerations. For example, we argue that in interactive drama, a DM shouldn't be seen as balancing externally imposed constraints with a player's freedom of action, but rather as a system that helps to ensure that there is enough narrative for the player to have a coherent and interesting experience.

Other experiences, however, may have genuinely external constraints that must be imposed in a way that could conflict with the user's freedom and goals. For example, a TTD-MDP based system was proposed for guiding museum tours [3]. In that domain, the goal of reducing congestion really is an external goal imposed on the visitors, and is reasonably expressed by targeting a specific distribution of experiences so as to keep visitors nicely spread out.

5. CONCLUSIONS

By separating the issues of what to optimize and how to carry out the optimization, we showed that the algorithms used by targeted trajectory Markov decision processes (TTD-MDPs) and by search-based drama management (SBDM) are versions of a generic search-based algorithm to which caching or offline computation may be added separately from the consideration of what to optimize.

On the conceptual issue, we defended the original formulation of drama management that sought to maximize an experience-quality function. We pointed out that experience-quality functions are not equivalent to story-quality functions that rate experiences as if they were non-interactive narratives, but are rather functions that explicitly take into account elements of a good interactive experience, such as the notion of interactive agency. We showed that TTD-MDPs, by contrast, primarily serve to add nondeterminism to their actions, which is a separate concern from interactive agency and does not necessarily produce agency.

6. REFERENCES

[1] J. Bates. Virtual reality, art, and entertainment. *Presence*, 1(1):133–138, 1992.

[2] S. Bhat, D. Roberts, M. J. Nelson, C. L. Isbell, and M. Mateas. A globally optimal algorithm for TTD-MDPs. In *Proceedings of AAMAS*, 2007.

[3] A. Cantino, D. L. Roberts, and C. L. Isbell. Autonomous nondeterministic tour guides: Improving quality of experience with TTD-MDPs. In *Proceedings of AAMAS*, 2007.

[4] B. Magerko. Story representation and interactive drama. In *Proceedings of AIIDE*, 2005.

[5] M. J. Nelson and M. Mateas. Search-based drama management in the interactive fiction Anchorhead. In *Proceedings of AIIDE*, 2005.

[6] M. O. Riedl, C. J. Saretto, and R. M. Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of AAMAS*, 2003.

[7] D. L. Roberts, S. Bhat, K. St. Clair, and C. L. Isbell. Authorial idioms for target distributions in TTD-MDPs. In *Proceedings of AAAI*, 2007.

[8] D. L. Roberts, M. J. Nelson, C. L. Isbell, M. Mateas, and M. L. Littman. Targeting specific distributions of trajectories in MDPs. In *Proceedings of AAAI*, 2006.

[9] P. Weyhrauch. *Guiding Interactive Drama*. PhD thesis, Carnegie Mellon University, 1997.

[10] R. M. Young, M. O. Riedl, M. Branly, A. Jhala, R. J. Martin, and C. J. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1(1), 2004.