Tractable POMDP Representations for Intelligent Tutoring Systems

JEREMIAH T. FOLSOM-KOVARIK, Soar Technology
GITA SUKTHANKAR, University of Central Florida
SAE SCHATZ, Mesh Solutions, LLC

With Partially Observable Markov Decision Processes (POMDPs), Intelligent Tutoring Systems (ITSs) can model individual learners from limited evidence and plan ahead despite uncertainty. However, POMDPs need appropriate representations to become tractable in ITSs that model many learner features, such as mastery of individual skills or the presence of specific misconceptions. This article describes two POMDP representations—*state queues* and *observation chains*—that take advantage of ITS task properties and let POMDPs scale to represent over 100 independent learner features. A real-world military training problem is given as one example. A human study ($n = 14$) provides initial validation for the model construction. Finally, evaluating the experimental representations with simulated students helps predict their impact on ITS performance. The compressed representations can model a wide range of simulated problems with instructional efficacy equal to lossless representations. With improved tractability, POMDP ITSs can accommodate more numerous or more detailed learner states and inputs.

Categories and Subject Descriptors: I.6.5 [**Simulation and Modeling**]: Model Development; K.3.1 [**Computing Milieux**]: Computers and Education

General Terms: Design, Performance, Experimentation

Additional Key Words and Phrases: Partially observable Markov decision processes, computer-based training, intelligent tutoring systems

ACM Reference Format:
Folsom-Kovarik, J. T., Sukthankar, G., and Schatz, S. 2013. Tractable POMDP representations for intelligent tutoring systems. ACM Trans. Intell. Syst. Technol. 4, 2, Article 29 (March 2013), 22 pages.
DOI = 10.1145/2438653.2438664 http://doi.acm.org/10.1145/2438653.2438664

This article is an extended version of Folsom-Kovarik, J. T., Sukthankar, G., Schatz, S., and Nicholson, D., "Scalable POMDPs for diagnosis and planning in intelligent tutoring systems," in *Proactive Assistive Agents: Papers from the AAAI Fall Symposium*, Association for the Advancement of Artificial Intelligence, Arlington, VA.

This work is supported in part by the Office of Naval Research Grant N0001408C0186, the Next-generation Expeditionary Warfare Intelligent Training (NEW-IT) program, and NSF award IIS-0845159. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Authors' addresses: J. T. Folsom-Kovarik (corresponding author), Soar Technology, 1942 West C. R. 419, Suite 1060, Oviedo, FL 32766; email: jeremiah.folsom-kovarik@soartech.com; G. Sukthankar, Department of Electrical Engineering and Computer Science, University of Central Florida, 4000 Central Florida Boulevard, Orlando, FL 32816; S. Schatz, Mesh Solutions, LLC, 12601 Research Parkway, Orlando, FL 32826.

DOI 10.1145/2438653.2438664 http://doi.acm.org/10.1145/2438653.2438664

1. INTRODUCTION

Intelligent Tutoring Systems (ITSs) are teaching or training programs that model individual learners' thought processes to estimate what kinds of help they need and adapt instruction so they learn more effectively. In the current state-of-the-art, several successful ITSs have used Bayesian inference principles to model learners at the levels of overall knowledge mastery or affective state estimation (e.g., Arroyo et al. [2006] and Conati et al. [2002]). Partially Observable Markov Decision Processes (POMDPs) are an important class of Bayesian planning model that hold promise for future intelligent tutors yet remain relatively unexplored. POMDPs describe both how to estimate reality from unreliable evidence and, additionally, how the actions a system can take might change the real world. POMDPs can plan ahead with action sequences that are optimal in the long term, while still maintaining flexibility to deal with changes in an uncertain environment [Kaelbling et al. 1998].

Uncertainty often affects intelligent tutors. Learners' mental processes and plans are often vague and difficult to discern [Chi et al. 1981]. Learners who need help can still make lucky guesses, while learners who know material well can make mistakes [Norman 1981]. Learner understanding is fluid and changes often. There are many possible reasons for mistakes, and it is difficult to exhaustively check all of them [VanLehn and Niu 2001]. Further, the effects of a particular intervention on a learner are rarely certain [Snow and Lohman 1984].

Despite the uncertainty inherent in teaching and training, human instructors and ITSs can still improve learning by successfully planning ahead. For example, a tutor might ask questions to confirm which underlying misconceptions caused an error, rather than simply correcting the most likely one [Graesser and Person 1994]. When a trainee makes a mistake during practice, a trainer might decide to delay feedback to avoid overload [Hattie and Timperley 2007]. Even encouraging a frustrated learner is an example of planning ahead because it takes time away from immediate instruction to make later instruction more effective. POMDPs are able to estimate the long-term impacts of their actions and plan interventions that improve learning over time [Kaelbling et al. 1998].

However, the price of POMDPs' power is their complexity. Modeling more learner states leads to exponential growth in memory and processing requirements. The scaling problems of large POMDPs probably help explain why existing POMDP ITSs are small, modeling fewer than 20 features such as "skill mastery" or "broad affective states" [Levchuk et al. 2013; Theocharous et al. 2009, 2010]. In contrast, ITSs deployed in classrooms may need to model over 100 of these features [Payne and Squibb 1990].

In order to create tractable POMDPs, researchers can make assumptions that limit the kinds of problems on which the POMDPs work well. The simplifying assumptions must exploit properties of the problems being represented, or they will cause poor performance; they must sacrifice unimportant information to keep important information. Certain properties of ITS tasks can make them easier to represent with POMDPs. The present article proposes two experimental POMDP representations to leverage these properties. *State queues* mitigate the state-space explosion that keeps POMDPs small, and *observation chains* increase the information a single assessment can convey to the ITS. Together, these representations let POMDPs model much larger problems, of the scale a real-world ITS would face.

Section 2 of this article contains background information about POMDPs, how they function, and how they relate to some established ITS architectures. Section 3 describes how state queues and observation chains can represent problems with properties common to ITS problems. Section 4 describes an initial study with human trainees and trainers that confirms the suggested model could represent a real-world training

problem. Section 5 describes three experiments with simulated students that test state queues and observation chains, showing that they perform well in a variety of conditions a real ITS might face.

2. PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

2.1. Definition

Tutoring can be modeled as a problem of sequential decision making under uncertainty. An ITS must select appropriate pedagogical actions such as corrections and hints, without having absolute knowledge of learners' mental states and without any guarantee that the actions will have the desired effect.

A Partially Observable Markov Decision Process (POMDP) is a general tool for modeling such problems. With a POMDP model, a system can infer underlying causes for observations and determine in advance how best to react to changes, even when observations and actions are unreliable [Kaelbling et al. 1998]. POMDPs have proven useful in non-ITS tasks such as controlling robots [Thrun et al. 2000], planning medical treatments [Hauskrecht and Fraser 2000], and interpreting spoken dialog [Young et al. 2010] or video [Hoey and Little 2007]. These applications highlight the potential power of POMDPs for finding long-term optimal actions in uncertain situations.

In general, a POMDP defines *states* that describe the real world at a given moment in time. States are discrete, finite, and mutually exclusive, but their true values are hidden from the system. A *belief* is a probability distribution over a set of states that estimates which states are more likely to be true. *Observations* provide evidence to update a belief. The true, hidden state of the real world makes certain observations more likely, though they are rarely perfect. In turn, system *actions* try to change the state of the real world. Action outcomes are not guaranteed. At any given time a *policy* selects the best action, depending on the current belief. Before execution, the POMDP parameters determine what policy will maximize an expected long-term *reward*. Rewards can be assigned to choosing beneficial actions or reaching goal states.

A particular POMDP is defined by a tuple $\langle S, A, T, R, \Omega, O \rangle$ comprising a set of hidden states S, a set of actions A, a set of transition probabilities T that define how actions change system states, a set of rewards R, a set of observations Ω, and a set of emission probabilities O that define which observations appear when the system is in a given state. Modeling a problem consists of defining these parameters. Then an algorithm can find a policy that consults the model and tells how to act [Kaelbling et al. 1998].

In summary, Figure 1 illustrates how the POMDP problem representation aligns with the processes and workflow many ITSs follow in modeling a learner, assessing and diagnosing the learner's needs, and choosing adaptive interventions. If S models the mental states of a particular learner, then A, T, and R model the effects of ITS interventions, connecting the POMDP with the ITS's pedagogical module. Likewise, Ω and O relate to ITS assessment and diagnosis modules. Thus, a POMDP learner model has the potential to integrate with more aspects of an ITS than many conventional learner models. The greater reach of the POMDP throughout the ITS adds to the model's future potential to build more intelligent interactions more efficiently with machine learning, although that potential will not be the subject of the present work.

2.2. Comparisons to Selected ITS Models

ITS practitioners familiar with other model architectures may find differences when considering POMDP models.

Compared to rule-based cognitive tutors, POMDPs operate at a higher level of abstraction. They have no need to specify neurological events (but no ability to leverage

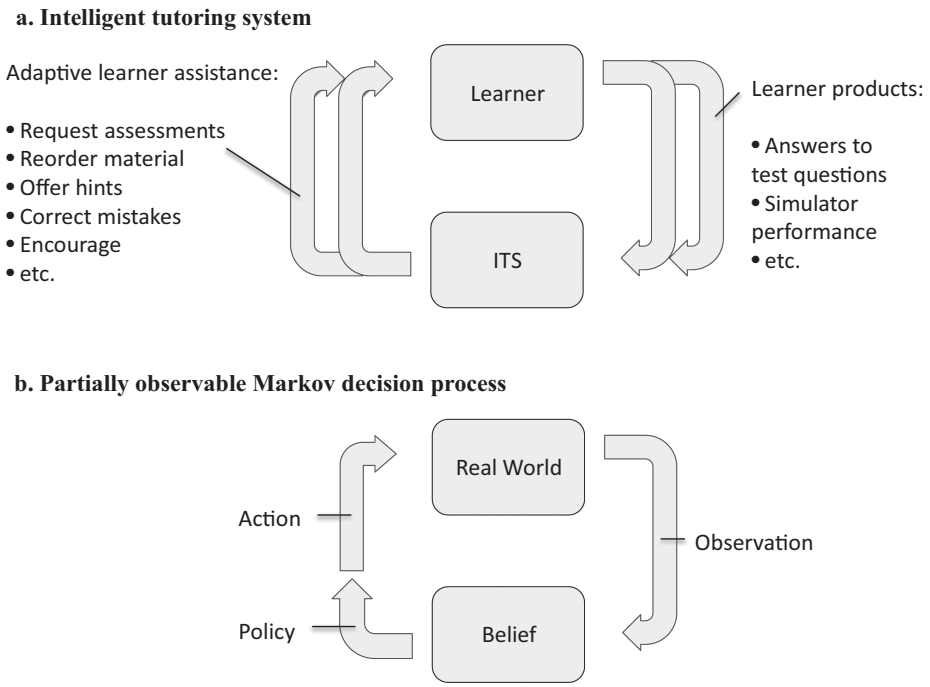

Fig. 1. The POMDP representation aligns with intelligent tutors' needs. Just as an ITS must model a learner's hidden mental states and adapt its interventions accordingly, a POMDP maintains a "belief" about the real world that is updated with each new observation and that determines optimal actions according to a predetermined policy.

them either). POMDPs are more closely related to knowledge tracing models, overlay-like structures that augment some cognitive tutors [Corbett and Anderson 1995] and can themselves be machine-learned Bayesian networks (e.g., Baker et al. [2010] and Ritter et al. [2009]). However, POMDPs may also model nonmastery states that affect learning.

POMDPs are also related to constraint-based tutors' long-term models [Mayo and Mitrovic 2001] and affective models [Zakharov et al. 2008] because both can make inferences about multiple states from a single assessment and can differentiate a misconception from a momentary slip. In a POMDP, the policy selects optimal actions according to Bayesian principles, but POMDP policies are typically determined only once while the constraint-based long-term model is updated during student interaction.

Manually constructed and machine-learned classifiers can also play the role of a learner model. Classifiers in ITSs typically sort individuals into broad groups based on mastery or cognitive traits. Examples of classifiers that have been used as learner models include decision trees (e.g., Cha et al. [2006] and McQuiggan et al. [2008]), neural networks (e.g., Castellano et al. [2007]), and ensemble methods (e.g., Hatzilygeroudis and Prentzas [2004] and Lee [2007]). POMDPs differ from methods that merely classify learners by integrating actions and action planning in the same model. Neural networks and very complex decision trees, such as ones that handle many uncertain situations, can be opaque, making decisions that are difficult to explain. In contrast, POMDPs can report their beliefs and recommendations in ways that users who are not expert in Bayesian principles understand [Almond et al. 2009].

In summary, POMDPs are capable of modeling learner information that is qualitatively similar to the data successful existing ITSs use. The similarity supports the intuition that a POMDP may effectively control an ITS, while the important differences represent interesting directions for future research into their consequences.

2.3. Comparisons to Bayesian Learner Models

POMDP ITSs build on the successes of other Bayesian models in intelligent tutors. Bayesian models can handle uncertainty and recover from errors, infer hidden state values from evidence, and allow machine learning of some model components during development. The usefulness of increasingly complex Bayesian models in ITSs suggests POMDPs could also control ITSs successfully.

As an initial example OLAE, an assessment tool and precursor to the Andes physics tutor, observes individual steps of learners' work to infer their domain knowledge mastery [VanLehn and Martin 1998]. Similarly, the ITS Ecolab observes learners' help use to predict both topic mastery and readiness to learn each new topic [Luckin and du Boulay 1999]. In both models, subject-matter experts designed networks by hand that used Bayesian methods to interpret assessments.

The Bayesian model architecture also lets developers apply machine learning to speed tutor development and refine accurate models from real learner data. For example, the high-school math tutor Wayang Outpost [Arroyo et al. 2004] observes help use to form mastery estimates. In this system, machine learning from experimental data helps define model parameters [Ferguson et al. 2006].

In the elementary grammar tutor CAPIT [Mayo and Mitrovic 2001], Bayesian methods inform a still larger portion of the tutor behaviors. Whereas previous Bayesian-model tutors selected pedagogical actions with heuristics or hand-written rules, CAPIT can predict action outcomes with its learner model. CAPIT also modifies its Bayesian model updates with a simple discount function to reflect the fact that learners' knowledge changes over time.

Dynamic Bayesian Networks (DBNs) are Bayesian networks that can model changes over time. An early version of the Andes physics tutor [Conati et al. 2002] used a DBN, and the Prime Climb math tutor [Conati 2002; Conati and Maclaren 2009] models learner goals and affective states with a similar, DBN-like construct that substitutes a non-Bayesian update method for better scaling.

Finally, a coin tutor for elementary students [Theocharous et al. 2010] uses a full-fledged POMDP learner model. For this prototype, problem state is assumed to be completely observable, concepts are ordered strictly linearly, and only one cognitive state is modeled (attention/distraction). Another instance of the tutor [Theocharous et al. 2009] has a six-state learner model with factorization [Boutilier et al. 1999] to address scaling problems, creating several small POMDPs that can be selected by a second, hierarchical POMDP. A POMDP that controls lesson selection in a military training setting has 11 states [Levchuk et al. 2013; Shebilske et al. 2009]. Compared to previous Bayesian ITSs, these POMDP ITSs model fewer facts about each learner.

In summary, Bayesian models allow diagnosis despite uncertainty and the possibility to base parameters and structure on expert requirements, empirical data, or both. Dynamic solutions such as DBNs and POMDPs can additionally model change over time. While DBNs estimate hidden states and require separate, hand-written rules to act on the diagnoses, optimal intervention planning is integral in POMDPs. However, existing POMDP ITSs only model relatively few features or learner states.

2.4. Tractability Limitations

POMDPs are characterized by important size limitations. Intelligent tutors often model numerous facts about each learner, for example, skill mastery or other cognitive states

that impact learning. Finding good policies and even storing a POMDP in memory can become difficult when more than a few of these features need to be modeled.

First, solving a POMDP to find a policy is exponentially complex according to the number of states modeled. Policies map beliefs to actions, and beliefs are probability distributions over states. Therefore, finding a good policy requires searching a hyperspace with as many continuous dimensions as the POMDP has states. Fortunately, approximate search algorithms exist, for example, ones that only evaluate selected points in belief space (e.g., Kurniawati et al. [2008], Lovejoy [1991], and Pineau et al. [2003]). However, memory and processor resource limitations still make policy search more difficult or even impossible as the number of states increases [Kurniawati et al. 2008; Poupart 2005].

Second, without any special representations, POMDP state counts grow exponentially with the number of independent variables or features to describe. For example, ITS designers might need to model ten binary facts about a learner. A naïve state representation would then require $|S| = 2^{10} = 1{,}024$ separate states to store all possible combinations of these values. Likewise, state transition probabilities, observation emission probabilities, and rewards all depend on system state and would therefore have exponentially increasing numbers of parameters to store (and either specify or discover empirically) in at least three more places, up to $|T| = |S|^2\ x\ |A|$, $|O| = |S|\ x\ |A|\ x\ |\Omega|$, and $|R| = |S|^2 x |A| x |\Omega|$. During the POMDP's policy learning and application, beliefs and policies would have similar space requirements.

There have been several general approaches to representing POMDPs more compactly. For more information, a good resource is Poupart [2005]. For example, state aggregation decreases the number of states in a POMDP by combining states that have the same reward values, even when they have different underlying meanings [Boutilier and Poole 1996; Feng and Hansen 2004; McCallum 2002]. To reduce the branching factor at each possible step in a policy, several atomic actions can be grouped together into macro-actions that dictate several turns in a row [Hauskrecht et al. 1998; He et al. 2010]. POMDP observations can be split into independent underlying components or aggregated into groups of observations that all have the same effect on planning, for example, because they give the same information about the current state [Hoey and Poupart 2005].

Although much effort has been dedicated to general methods for alleviating POMDPs' exponential costs, general methods must be able to handle possibilities that may never apply in a particular problem domain. An important way to make a POMDP tractable is to focus on a single problem class and take advantage of regularities in that class. The present article describes two POMDP representations that can efficiently model the process of intelligent tutoring.

3. MAKING POMDPS PRACTICAL FOR ITS APPLICATIONS

This section describes two compression schemes that make POMDPs capable of representing problems as large as the ones real-world ITSs confront. The first, state queues, compresses a POMDP's representation of system states, while the second, observation chains, compresses a POMDP's observation representation. Both compression schemes use properties of the tutoring task to discard some data for better scaling. The specialized representations work with generic search algorithms to find effective policies for large problems.

State queues and observation chains are designed to compress a wide variety of possible problem representations. The following subsection introduces a simple example of a learner model, M, with a specific state, action, and observation structure. The model M has the same structure as the one that produced the empirical results in this

article. The components in the example model can, in turn, be assigned any number of interpretations in specific instructional domains.

3.1. An Example POMDP Learner Model

POMDP hidden states can model the current mental status of a particular learner at the level of *knowledge states* and *cognitive states*. Knowledge states describe tutee knowledge or skill mastery, while cognitive states describe any nonmastery states that affect learning.

In the example learner model, K is the domain-specific set of specific misconceptions or facts or competencies the learner has not grasped. By itself, K is related to a buggy or perturbation model, learner models with which ITSs have successfully detected common misconceptions or malrules in novices (e.g., Johnson [1990] and Shute et al. [2008]). In the present article, members of K are called *gaps* to indicate that each element can represent either the presence of a misconception or the absence of a required mastery. The intelligent tutor should act to discover and remove all gaps.

In addition to the knowledge states in K, the learner state model S also contains cognitive states. The set C of cognitive states represents transient or permanent properties of a specific learner that may change action efficacy. Cognitive states can also explain observations. When possible, the intelligent tutor should account for a learner's cognitive state so as to remove gaps more effectively. Cognitive states are not domain specific, although the instructional domain might determine which states are important to track. In the example learner model, the set C includes boredom, confusion, frustration, and flow.

Many hints or other actions could tutor any particular gap. Each POMDP action i has a chance to correct each gap j with base probability a_{ij}. The interventions available to the ITS determine the values of a, and they must be either fitted from empirical data or set by a subject-matter expert. In M, $A = K\ U\ \{noop\}$, and POMDP actions decide which gap to address (or none). A pedagogical module, separate from the POMDP, is posited to choose an intervention that tutors the target gap effectively in the current context. A POMDP could also control ITS actions directly, with no intervening pedagogical module, but a values would be needed for each action.

The learner's cognitive state may further increase or decrease an action's ability to correct a gap according to a modifying function f_c: [0,1]→[0,1]. In M, values for f_c approximate trends empirically observed during tutoring [Craig et al. 2004]. Transitions between the cognitive states are also based on empirical data [Baker et al. 2007; D'Mello et al. 2007]. Actions which do not succeed in improving the knowledge state have higher probability to trigger a negative affective state [Robison et al. 2009]. Because of the cognitive state modifications in f_c, an ITS might improve its overall performance by not intervening immediately, such as when there is too much uncertainty about a learner's state.

For simplicity, no actions in M introduce new gaps. In the present experiment's instructional domain, training takes place over a short time period so forgetting learned information is less likely. However, in many domains it is not realistic to assume actions cannot create gaps, so future work should relax this assumption by adding transitions in M that model misunderstanding or forgetting. The extra transitions will increase the number of parameters to define, but not change M qualitatively.

Finally, an external assessment module is also posited to preprocess ITS observations of learner behavior into POMDP observations of assessment lists. Thus, observations in M are (possibly noisy) indicators that particular gaps are present or absent. Each observation can contain information about multiple gaps. The POMDP's task is to find patterns in these assessments and transform them into a coherent diagnosis in its belief state. For simplicity, M does not include direct observations of cognitive states.

3.2. Tutoring Problem Characteristics

Often, tutoring problems have four characteristics that can be exploited to compress their state-space representations. These are a natural ordering on states, few action side-effects, state independence, and feature-rich observations.

First, in many cases instructors naturally address knowledge gaps or misconceptions in a particular order. Often, certain gaps are difficult to address in the presence of other gaps. For example, when a person learns algebra it is difficult to understand exponentiation before multiplication. Multiplication in turn is difficult to comprehend without a grasp of addition. This relationship property allows instructors in general to assign a partial ordering over the misconceptions or gaps they wish to address. They make sure learners grasp the fundamentals before introducing topics that progressively build on the learners' knowledge.

To reflect the way the presence of one gap, k, can change the possibility of removing another gap j, a final term $d_{jk} : [0, 1]$ is added to M. The probability that an action i will clear gap j when any other gaps k are present then becomes $f_c(a_{ij}\Pi_{k,k\neq j}(1 - d_{jk}))$. Values of d_{jk} near 1 indicate that for pedagogical reasons gap j "depends on" gap k, and it is difficult to remove gap j as long as gap k exists.

Second, in tutoring problems each action often affects a small proportion of K. Presenting a lesson about addition will not give a student a sudden understanding of multiplication. Furthermore, in ITS model design, interventions often target individual gaps. For example, an ITS that models a specific misconception about carrying during addition is likely to contain an intervention targeting that misconception. In contrast, an ITS that models several addition misconceptions but addresses all of them by reteaching the entire addition lesson does not meet this criterion. The second characteristic holds true for ITS problems where $a_{ij} = 0$ for relatively many combinations of i and j.

Third, the presence or absence of knowledge gaps in the initial knowledge state can be close to independent, that is, with approximately uniform co-occurrence probabilities. Finding that a learner has a misconception about carrying in addition does not make it more or less likely the learner will need drilling in multiplication facts. This characteristic is the most restrictive, in that it is probably less common in the set of all tutoring problems than the other three, but such problems do exist. For example, in cases when a tutor is teaching new material, all knowledge gaps will be likely to exist initially, satisfying the uniform co-occurrence property.

Finally, ITSs often receive rich inputs that include evidence about several orthogonal features in a single observation. For example, a learner's success on a math problem that requires addition, multiplication, and exponentiation could form evidence for mastery of all the competencies needed to complete the task. Conversely, specific kinds of mistakes could be evidence the student has mastered some math skills but certain misconceptions still exist. The various skill assessments represent independent dimensions that are all transmitted to the ITS by observing a single event.

Together, these four characteristics of teaching and training enable the *state queue* and *observation chain* POMDP representations to represent the ITS problem tractably.

3.3. State Queues

In tutoring problems, a partial ordering over K exists and can be discovered, for example, by interviewing subject-matter experts. By reordering the members of K to minimize the values in d where $j < k$ and breaking ties arbitrarily, it is further possible to choose a strict total ordering over the knowledge states, or priority. This ordering does not necessarily correspond to the POMDP's optimal action sequence, because of other considerations such as current cognitive states, initial gap likelihoods, or action

efficacies. However, it gives a heuristic for reducing the number of states a POMDP must track at once.

A *state queue* is an alternative knowledge state representation. Rather than maintain beliefs about the presence or absence of all knowledge gaps at once, a state queue only maintains a belief about the presence or absence of one gap, the one with the highest priority. Gaps with lower priority are assumed to be present with the initial probability until the queue reaches their turn, and POMDP observations of their presence or absence are ignored except insofar as they provide evidence about the priority gap. POMDP actions attempt to move down the queue to lower-priority states until reaching a terminal state with no gaps.

The state space in a POMDP using a state queue is $S = C \times (KU \{done\})$, with *done* a state representing the absence of all gaps. Whereas an enumerated state representation grows exponentially with the number of knowledge states to tutor, a state queue grows only linearly.

State queuing places tight constraints on POMDPs. However, these constraints may lead to approximately equivalent policies and outcomes on ITS problems, with which they align. If ITSs can only tutor one gap at a time and actions only change the belief in one dimension at a time, it may be possible to ignore information about dimensions besides the one to move in first. Finally, the highly informative observations that are possible in the ITS domain may ameliorate the possibility of discarding some.

3.4. Observation Chains

When emission probabilities of different dimensions are independent, conditioned on the underlying state, one observation can be serialized into multiple observations that each contains nonorthogonal information. A process for accomplishing this is *observation chaining*. Under observation chaining, observations decompose into observation chain elements. Decompositions are chosen so that emission probabilities within chain elements are preserved, but emission probabilities between elements are independent.

As an example, imagine a subset of Ω that informs a POMDP about the values of two independent binary variables, X and Y. To report these values in one observation, Ω might contain four observations $\{XY, X'Y, XY', X'Y'\}$. An observation chain would replace these with four chain elements, $\{X, X', Y, Y'\}$. Then where one original observation would appear, a chain of two equivalent components is observed instead. Whenever one observation completely describes more than two independent features, a chain representation will add fewer observations to Ω than an enumerated one.

Chains can be of arbitrary length, so a token end is added in Ω to signal the end of a chain. In M, the simplified example learner model, assessments a POMDP observes can signal the presence of a particular knowledge gap, its absence, or neither. A *blank* chain element that conveys no information signals an observation that is uninformative about any gap. With observation chaining, $\Omega = K \cup \{blank\}) \cup (K' \cup \{blank\}') \cup \{end\}$.

In POMDPs with observation chaining, S contains an unlocked and a locked version of each state. Observing the end token moves the system into an unlocked state, while any other observation sets the equivalent locked state. Transitions from unlocked states are unchanged, but locked states always transition back to themselves. Therefore, the ITS can observe any number of chain elements before it recommends another action, and the POMDP controller does not need to be rewritten. Observation chains make Ω scale better with the number of orthogonal observation features, at the cost of doubling $|S|$.

Observation chains are similar to the plan Hoey and Poupart [2005] suggest for reducing conditionally independent observation dimensionality. However, where they add a counter to the state space and transmit observations in order, the lock-unlock scheme does not impose an order on chain elements and requires no counter. Therefore,

reducing observation space does not require increasing state space by the same factor, as Hoey and Poupart's [2005] algorithm does. Instead, observation chains deliver exponential observation-space improvements with only a one-time doubling of state space cardinality.

3.5. Representation Summary

This section describes an example learner model structure that lets a POMDP represent an adaptive tutoring problem, and two compression schemes that make large problems possible to represent with such structures. The information discarded during compression is theorized to have little impact on outcomes for certain classes of real-world problems, including the intelligent tutoring problem. Section 5 describes experiments designed to explore situations under which the compression leads to effective tutoring.

4. HUMAN STUDY

A specific instructional domain was studied in order to illustrate how POMDPs can represent ITS problems. The domain, a training scenario that is currently in use by the U.S. military, grounded the ITS-specific POMDP representations in a real-world example and helped explore any assumptions about ITS tasks and the corresponding POMDP structures.

This section describes a study of the instructional domain with a human trainer and trainees ($n = 14$). The study was designed to determine which states an ITS for the domain should model. Similar studies with larger sample sizes could also yield information about actions, observations, and rewards in a particular domain. The study also gathered information about different model input channels, which will be reported elsewhere. The study results provided preliminary validation of the proposed POMDP learner model.

4.1. The CFF Instructional Domain

The training task, known as Call For Fire (CFF), is performed by United States Marine Corps personnel called Forward Observers (FOs). FOs let distant artillery and other units engage targets with precise fire that is more likely to be effective and less likely to harm friendly units or noncombatants in the area.

Trainees in the CFF domain must locate potential targets, identify which are foes, prioritize threats, and transmit target descriptions (themselves called CFFs) to the artillery. Each CFF transmission must contain accurate details about a target's position, its type, one of two ammunition types to use, and one of two firing patterns to use. CFF errors can have different underlying causes, such as either incorrectly identifying a unit's type or misremembering the prescribed method to attack that unit type.

In the U.S. Marine Corps, FOs may train to perform the CFF task with the Deployable Virtual Training Environment (DVTE), a laptop-based suite of training programs [Bailey and Armstrong 2002]. The DVTE program that trains the CFF task is a single-user, first-person simulation controlled with a mouse and keyboard. In this simulation, the role of the artillery is played by the computer.

Studying the CFF instructional domain helps ensure that the present work formulates an ITS problem realistically. Any POMDP that can usefully represent training in the CFF domain has successfully scaled to represent a real-world ITS problem.

4.2. Method

Undergraduate volunteers were trained to perform the CFF task in the DVTE. Each training and testing session took 2.5 hours. The course included training on using the simulator (Practice 1) and on the domain task (Practice 2), a test on the domain task

(Test1), and a test on a transfer task requiring similar skills in a new setting (Test 2). In previous studies using the same CFF task, material in Practice 2 was normally broken into two or more scenarios [Vogel-Walcutt et al. 2009, 2008]. The present study combined these scenarios to introduce more material at once, for the purpose of degrading trainees' support and causing them to display more misconceptions.

Trainees' performance was measured in time needed to generate a CFF, number of CFFs that resulted in a hit, errors in CFF descriptions or details, and target prioritization. Final performance was used as a proxy for learning because, based on intake forms, all trainees initially had no knowledge of the CFF task. It was not possible to administer a pretest for the same reason, and in addition pretests in this domain can act as advance organizers, unduly influencing training outcomes.

During training and testing, trainees were encouraged to share their thoughts in real time, to help reveal their knowledge and cognitive states to researchers. In one condition, participants ($n = 7$) were encouraged to ask questions of the trainer. In a second condition, participants ($n = 7$) were required to "think aloud" [Ericsson and Simon 1984; Fonteyn et al. 1993], describing the conscious thought processes underlying their actions in the simulator. The differing conditions were part of an ongoing effort to explore the role of learner questions in updating a cognitive model, but in the present work they will not be considered separately. In both conditions, a human trainer answered all questions.

The cognitive and knowledge states trainees displayed were observed and recorded by two researchers. Based on input from subject-matter experts and pilot studies, 31 knowledge states were initially hypothesized to comprise a sufficient model of the misconceptions and gaps trainees could have. In addition, several cognitive states were explored. After each simulator interaction, trainees reported how often they had felt each of six affective states on a seven-point Likert scale that was created for the present study. The states were the ones studied by Craig et al. [2004], Baker et al. [2007], Rodrigo et al. [2007], and D'Mello et al. [2007]: boredom, confusion, flow, frustration, delight, and surprise. In accordance with that work, the first four states named were hypothesized to appear more often during training than the last two. The states were functionally defined for the participants as in D'Mello et al. [2006] and Graesser et al. [2006]. Retrospective trainee reports of mental effort as measured by the Cognitive Load Questionnaire [Paas 1992] were also collected after each simulator and nonsimulator task.

4.3. Results and Discussion

The average number of investigator observations per trainee was 33.25 ($\sigma = 10.2$). There were also 42 cognitive state self-reports collected during each training session. Together these represented a quantitative and qualitative improvement over the learner data usually available from simulator performance metrics in this domain (inputs per trainee $\mu = 25.85, \sigma = 2.32$). The additional data helped explore the knowledge states and cognitive states trainees are likely to experience during training in the CFF domain.

As Figure 2 shows, the CFF training conducted by the human trainers succeeded in making trainee CFFs faster and more accurate after practice. Since training was effective according to several measures, the knowledge gaps and cognitive states investigators observed during the study are likely to comprise the most important or most frequently occurring components of a useful ITS model for this domain and user population.

Of the 31 misconceptions or other knowledge gaps learners were expected to experience during training, investigators actually observed 25 during the human study. In addition, two new domain knowledge gaps and three new gaps relating to simulator

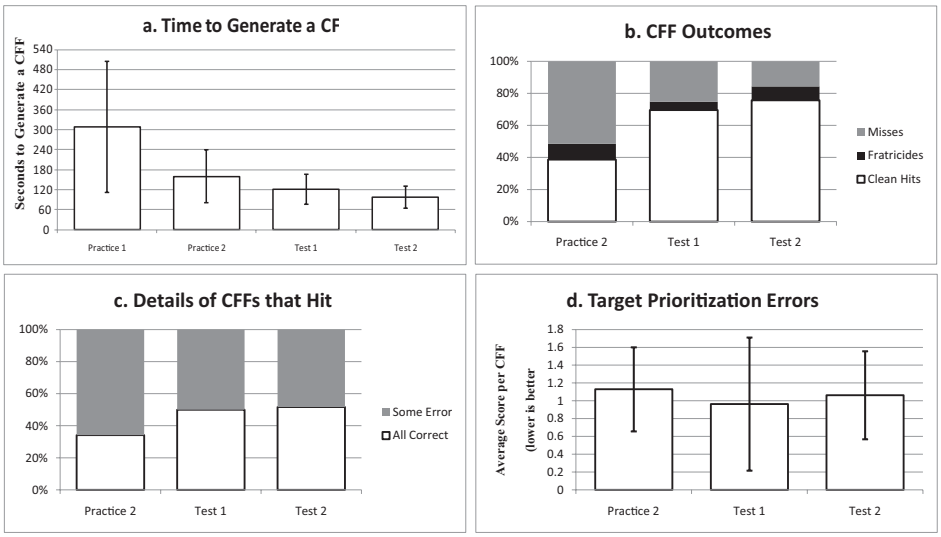

Fig. 2. Trainee performance improved on several measures during the human study. Time to generate a CFF decreased (a) while the number of CFFs that resulted in hits and the percentage of hits with no errors increased (b, c). Target prioritization improved slightly, not significantly (d).

usage were observed at least once. These will be evaluated for incorporation into the learner model. Overall, the participants' experiences appeared to fit within the approximate number and detail level of the hypothesized knowledge states, suggesting that they are sufficient to model training in the CFF domain.

Like the hypothesized knowledge states, the human study also supported the selection of the hypothesized cognitive states. The most commonly reported cognitive states were flow (94% of reports in the mid or high range) and delight (79%). The reported occurrence of delight was both higher and more varied than hypothesized, suggesting its usefulness for a learner model if its impact on training effectiveness can be quantified. A reason for the difference might be that participants were actually reporting satisfaction, a state functionally similar to delight but with lower magnitude [Graesser et al. 2006]. The other hypothesized cognitive states also appeared often enough to allow modeling them: confusion (45%), surprise (42%), boredom (39%), and frustration (37%).

Self-reports of cognitive load showed little variability. Participants reported median values of cognitive load in most cases (63%). Instances of reporting cognitive load as high (24%) or low (13%) did not appear to correlate with any task or with trainee performance. The overall low variability of the cognitive load measure during this study suggests that it is less useful to model during CFF training. However, the usefulness of modeling cognitive load is likely to increase at more detailed timescales than this study recorded, such as second-to-second changes.

To summarize, the human study of CFF training supported incorporating the hypothesized knowledge states and cognitive states into an ITS learner model for the domain. Participants' experiences also suggested new knowledge gaps to incorporate as model states. Additional data collection would yield information about more model parameters such as state transitions, but for the purpose of the current study these parameters will be set by subject-matter experts instead. The following section combines the selected states with various model structures and parameters in a simulated training environment to explore the model's viability and sensitivity to parameter settings.

5. SIMULATION EXPERIMENTS

Three experiments were conducted on simulated students to empirically evaluate ITSs driven by different POMDP representations of the same underlying learner model. Empirically testing the performance of different representations helped estimate how the experimental compression schemes might impact real ITS learning outcomes.

5.1. Method

POMDPs and their accompanying policies represented ITSs and were presented with simulated students. In each simulation, a student was first generated with a random set of knowledge gaps and a random initial cognitive state from among those modeled. An ITS then had a limited number of actions to tutor the student and remove as many knowledge gaps as possible. Given a student's hidden mental reality, each ITS action had a specific probability to clear gaps, change cognitive states, and generate new student output observations that let the ITS update its plan. Changes in the students' mental states always occurred according to the probabilities specified in the model M described in Section 3. Therefore, M was an exact model of a simulated student, and the simulation experiments tested the experimental POMDPs' ability to represent M usefully.

Simulations tested a range of models with the M structure, to correspond to multiple real-world instructional domains. Specific parameter values appear in Section 5.2.

The first experiment was designed to explore the effects of problem size on the experimental representation loss during tutoring. *Lossless* POMDPs were created that used traditional, enumerated state and observation encodings and preserved all information from M, perfectly reflecting the simulated students. *Experimental* POMDPs used either a state queue or observation chains, or both. For problems with $|K| = |\Omega| = 1$, the experimental representations do not lose any information. As K or Ω grow, the experimental POMDPs compress the information in their states and/or observations, incurring an information loss and possibly a performance degradation.

Since the lossless representation had a perfect model of ground truth, the initial hypothesis in this experiment was that an ITS based on the experimental representations would produce student scores that differ from a lossless representation ITS by no more than 0.25 standard deviations (Glass' delta). This threshold was chosen to align with the Institute of Education Science's standards for indentifying substantive effects in educational studies with human subjects [U. S. Department of Education 2008]. A difference of less than 0.25 standard deviations, though it may be statistically significant, is not considered a substantive difference.

The second experiment measured the sensitivity of the experimental representations to the tutoring problem parameters a and d. Testing several parameter combinations helped understand how well the experimental representations could model a variety of instructional settings or other planning problems that fulfill the underlying assumptions.

As introduced in Section 3, d is a parameter describing the difficulty of tutoring one concept before another. When d is low, the concepts may be tutored in any order. Since a state-queue POMDP is constrained in the order it tries to tutor concepts, it might underperform a lossless POMDP on problems where d is low. Furthermore, the efficacy of different tutoring actions can vary widely in real-world problems. Efficacy of all actions depends on a, including those not subject to ordering effects.

Simulated tutoring problems with a in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and d in $\{0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.50, 0.75, 1.00\}$ were evaluated. To maximize performance differences, the largest possible problem size was used, $|K| = 8$. In problems with more than eight knowledge states, lossless representations were not able to learn

useful policies under some parameter values before exhausting all eight gigabytes of available memory.

The experimental representations were hypothesized to perform no more than 0.25 standard deviations worse than the lossless representations. Parameter settings that caused performance to degrade by more than this limit would indicate problem classes for which the experimental representations would be less appropriate in the real world.

The third experiment explored the performance impact of POMDP observations that contain information about more than one gap. Tests on large problems ($|K| \geq 16$) showed the experimental ITSs successfully tutor fewer gaps per turn as size increases. One cause of performance degradation on large problems was conjectured to be the information loss associated with state queuing. As state queues grow longer, the probability increases that a given observation's limited information describes only states that are not the priority state. If the information loss with large K impacts performance, state queues may be unsuitable for representing some real-world tutoring problems that contain up to a hundred separate concepts (e.g., Payne and Squibb [1990]).

To assess the feasibility of improving large problem performance, the third experiment varied the number of gaps that could be diagnosed based on a single observation. Observations were generated, by the same method used in the other experiments, to describe n equally sized partitions of K. When $n = 1$, observations were identical to the previous experiments. With increasing n, each observation contained information about the presence or absence of more gaps the ITS should tutor. Observations with high-dimensional information were possible to encode with observation chaining. This experiment would not conclusively prove that low-information observations degrade state queue performance, but would explore whether performance could be improved in high-information settings.

5.2. Experimental Setup

ITS performance in all experiments was evaluated by the number of gaps remaining after t ITS-tutee interactions, including no-op actions. In these experiments, the value $t = |K|$ was chosen to increase differentiation between conditions, giving enough time for a competent ITS to accomplish some tutoring but not to finish in every case.

All gaps had a 50% initial probability of being present. Any combination of gaps was equally likely, except that starting with no gaps was disallowed. Simulated students had a 25% probability of starting in each cognitive state.

The set of ITS actions included one action to tutor each gap, and a no-op action. Values for a and d were $a_{ij} = 0.5$ where $i = j$, and 0 otherwise; and $d_{ij} = 0.5$ where $i < j$, and 0 otherwise. These values were hypothesized to be moderate enough to avoid extremely good or bad performance, and reasonable for representing at least some real-world tutoring tasks. Nonzero values varied in the second experiment.

POMDP rewards were used only for policy search, not for evaluating POMDP performance. The reward structure was necessarily different for lossless and state-queue POMDPs because state queues do not model some knowledge gap information. However, both reward structures emphasized eliminating gaps as quickly as possible. For *lossless* POMDPs, any transition from a state with at least one gap to a state with j fewer gaps earned a reward of $100\ j/|K|$. For *state-queue* POMDPs, any transition that changed the priority gap from a higher-priority i to a lower-priority $i - j$ earned a reward of $100\ j/|K|$, but resolving any gaps other than the priority gap was not rewarded. In both conditions, changes in cognitive state did not earn any reward. The reward discount was 0.90. Goal states were not fully observable.

Policy search was conducted with the algorithm Successive Approximations of the Reachable Space under Optimal Policies (SARSOP). SARSOP is a point-based search

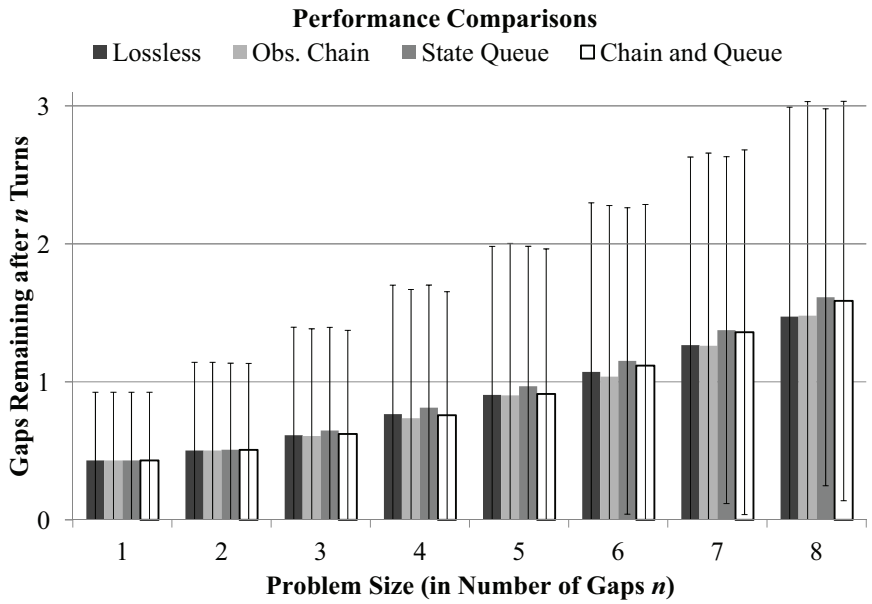

Fig. 3. Direct comparison to a lossless POMDP on problems with |K| $\leq$ 8, a = 0.5, d = 0.5 shows the experimental representations do not substantively degrade tutoring performance.

algorithm that quickly explores large belief spaces by focusing on the points that are more likely to be reached under an approximately optimal policy [Kurniawati et al. 2008]. SARSOP's ability to find approximate solutions to POMDPs with many hidden states makes it suitable for solving ITS problems, especially with the large baseline lossless POMDPs. Version 0.91 of the SARSOP package was used in the present experiment [Wei 2010].

5.3. Results

The first experiment did not find substantive performance differences between experimental representations and the lossless baseline. Figure 3 shows absolute performance degradation was small in all problems. Degradation did tend to increase with $|K|$. Observation chaining alone did not cause any degradation except when $|K| = 8$. State queues, alone or with observation chains, degraded performance by less than 10% of a standard deviation for all values of $|K|$.

Lossless POMDPs were not able to represent problems with $|K| > 8$. However, extrapolating from the results of the first experiment suggested that if a lossless POMDP with larger memory and processor resources could represent a problem with $|K| = 16$, its performance would be more than 0.25 standard deviations better than the experimental representations. This extrapolation motivated the third experiment, which tested the extent to which more informative observations could mitigate performance degradation on large problems.

The second experiment tested for performance differences under extreme values of action efficacy a and priority effect d. In lossless POMDPs, performance depended mostly on action efficacy. Varying the strength of the priority effect caused relatively small performance differences. Furthermore, observation chaining POMDPs performed substantively the same as lossless POMDPs under every combination of parameters.

Differences did appear for POMDPs with a state queue alone or a state queue combined with observation chaining. With a state queue alone, POMDPs performed

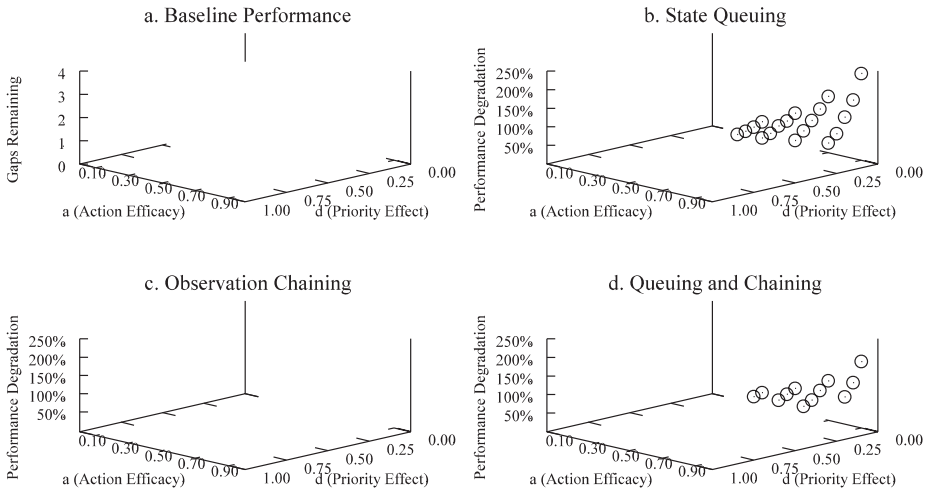

Fig. 4. When |K| = 8, |A| = 8, and |Ω| = 64, as here, lossless POMDP performance (a) varies with action efficacy, not priority effect. Experimental POMDPs perform about the same as lossless in many conditions (b, c, d), but state queues substantively degrade performance at some points (circled), indicating they are unsuitable to represent problems with priority effect d < 0.25.

substantively worse than the lossless baseline on tutoring problems with $d < 0.25$. The relative degradation increased on problems with greater action efficacy because the lossless POMDPs' better performance magnified small absolute differences. With a state queue and observation chaining combined, POMDPs performed substantively worse on problems with $d < 0.20$. The improvement over queues alone may be attributable to more efficient policy learning possible with smaller Ω.

Figure 4 shows the performance degradation in the second experiment POMDP using both state queues and observation chains, as a percentage of the lossless performance. Performance changes that were not substantive, but were close to random noise, ranged from 0% to 50% worse than the lossless representation. The first substantive difference came at $d = 0.15$, with a 53% degradation. The most substantive difference was 179% worse, but represented an absolute difference of only 0.22 (an average of 0.34 gaps remaining, compared to 0.12 with a lossless POMDP).

The third experiment (see Figure 5) showed the effect on tutoring performance of observations with more information. For $|K| = 32$, tutoring left an average of 9.4 gaps when each observation had information about one gap, and 3.7 when each observation described 16 gaps. For $|K| = 64$, the number of gaps remaining improved from 21.5 to 9.2. Finally, when $|K| = 128$, average performance increased from leaving 47.9 gaps to leaving just 22.3 with information about 32 gaps in each observation. However, for this size problem, doubling the available information again did not further improve performance.

5.4. Discussion

Together, the simulated experiments in this section show encouraging results for the practicality of POMDP ITSs.

The first experiment demonstrated that information compression in the experimental representations did not lead to substantively worse performance for problems small enough to directly compare. The size limit on lossless representations ($|K| \leq 8$) is too restrictive for many ITS applications (e.g., Payne and Squibb [1990]). The limit was caused by memory requirements of policy search. Parameters chosen to

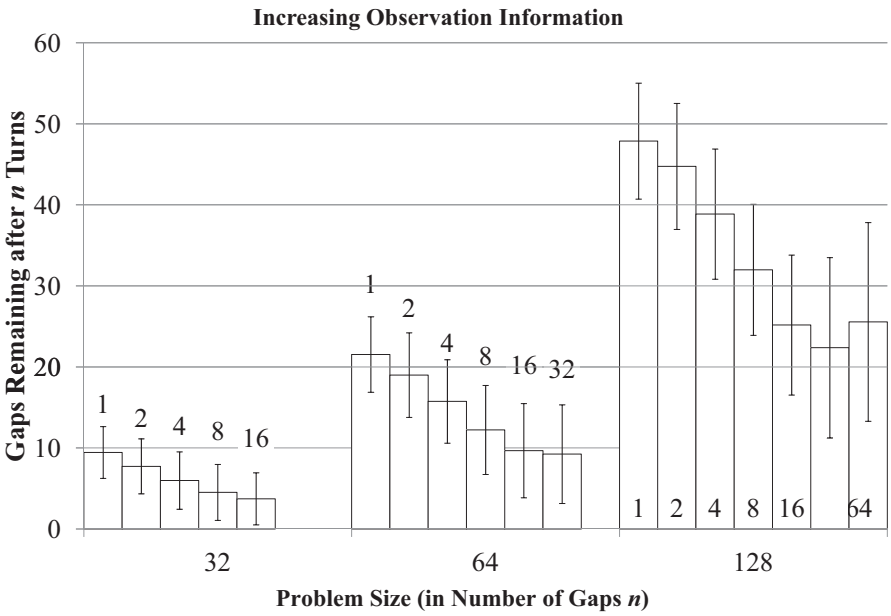

Fig. 5. Performance on large problems can be improved up to approximately double, if every observation contains information about multiple features. Observation chaining greatly increases the number of independent dimensions a single observation can completely describe (the number above each column).

shorten policy search could increase maximum problem size by only one or two more. Furthermore, one performance advantage of using enumerated representations stems from their ability to encode complex relationships between states or observations. But the improvement may not be useful if it is impractical to learn or specify each of the thousands of relationships.

The second experiment suggested that although some types of problems are unsuitable for the experimental representations, problems with certain characteristics are probably suitable. Whenever gap priority had an effect on the probability of clearing gaps $d \geq 0.2$, state queues could discard large amounts of information (as they do) and retain substantively the same final outcomes. Informal discussions with subject-matter experts suggest many real-world tutoring problems have priorities near $d \approx 0.5$.

Although it would be premature to compare simulation results to the performance of any ITS on real learners, there is a way to provide context for the value of a POMDP in tutoring the simulated problems. Figure 6 compares the results of the second experiment to a simulated *reactive ITS* that does not model knowledge gaps, but simply tutors the last gap observed. The comparison shows that, unlike POMDPs, the reactive ITS's performance was affected by the priority effect d. Figure 6(b) displays the gaps the POMDP ITS left remaining as a percentage of the reactive ITS's gap counts. The POMDP performed substantively better in almost all cases, achieving learning improvements between 0.09 and 0.72 standard deviations.

The third experiment demonstrated one method to address performance degradation in large problems. Adding more assessment information to each observation empirically improved performance. Observing many dimensions at once is practical for a subset of real-world ITS problems. For example, it may be realistic to assess any tutee performance, good or bad, as demonstrating a grasp of all fundamental skills that lead up to that point. In such a case one observation could realistically rule out many gaps.

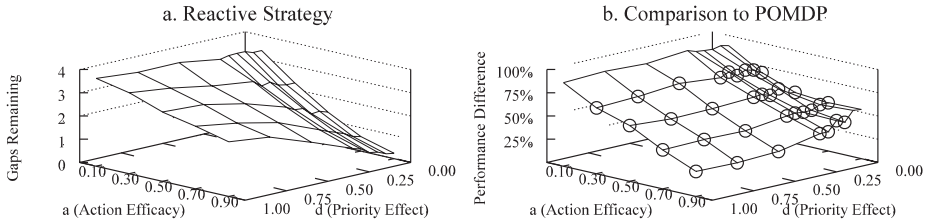

Fig. 6. Simply tutoring the most recently observed gap (a) causes poor performance when gap priority is important. POMDP scores. (See Figure 4(a)) are a fraction of the comparison scores at all points (b). The POMDP performs substantively better at many points (circled).

In the third experiment, problems with $|K| = 128$ and more than 32 gaps described in each observation did not finish policy learning before reaching memory limits. This suggests an approximate upper limit on problem size for the experimental representations. POMDPs that are able to encompass 128 gaps (1,024 POMDP states) could be sufficient to control ITSs that tutor many misconceptions, such as Payne and Squibb [1990]. Even larger material could be factored to fit within the limit by dividing it into chapters or tutoring sessions.

In summary, the three simulation experiments suggest that POMDPs compressed with state queues and observation chains are promising for use on ITS problems when topics must be taught in order. The Call For Fire (CFF) domain and other domains where model structure reflects instructional material ordering and dependencies (e.g., Conati et al. [2002] and Luckin and du Boulay [1999]) align well with POMDP ITSs. The structures may be less appropriate when topics can appear in many orders, such as in exploratory learning environments (e.g., Shute and Glaser [1990]). Second, although state queues were shown in the present work to accommodate state counts that can model many real ITS problems, they are still not appropriate for very large numbers of states, such as in highly detailed moment-to-moment models like production rule systems use (e.g., Anderson [1993]). Finally, observation chains are helpful when a large observation space can be factored into several conditionally independent observations. An assessment module is sufficient but not necessary to create such observations. However, observation chains are not useful if a large number of observations must relate directly to states through a complex function that cannot be factored, and then a different representation must be used.

The representations discussed in this article have advantages in some circumstances over other POMDP simplifications. Compared to existing examples of state-factored or hierarchical POMDP ITSs [Theocharous et al. 2009, 2010], state queues better preserve relationships between cognitive states or other modes that form the dividing lines in factored POMDPs, at the cost of some knowledge state information. State queues can be used in conjunction with other state factoring schemes. Compared to macro-actions that shorten lookahead horizons by grouping actions [He et al. 2010], the compressed representations do not limit observations or constrain the range of tutoring options. While macro-actions may be unnecessary with the present compressed representations which also make policy search easier, they can be used together. Finally, an alternative observation compression scheme is the one proposed by Hoey and Poupart [2005], in which all observations that lead to the same policy are aggregated. This scheme can integrate with the smaller state spaces of state queues, in place of observation chains. However, that algorithm requires data to learn the aggregation and therefore works best with a small number of possible plans, whereas ITSs usually have many actions which can be combined into even more possible plans, making compression schemes that do not take advantage of domain knowledge difficult to implement.

6. CONCLUSIONS

This article discussed applying POMDPs in intelligent tutors. With POMDPs, intelligent tutors can plan action sequences despite uncertainty. One sample method for encoding the ITS problem in a POMDP was described which can expand to include many knowledge states and cognitive states that existing ITSs model. The article studied a real-world training problem and its knowledge state characteristics. POMDPs are suitable for representing this real-world problem.

POMDPs are known to scale poorly with naïve problem representations. This article introduced two representations, state queues and observation chains, that let POMDPs represent large problems of the size ITSs might reasonably face, while staying small enough to find an effective policy. State queues and observation chains were studied with simulated students to determine their impact on ITS performance. Although the compressed representations are not suitable for some tutoring problems, they can represent a wide range of problems without damaging instructional efficacy.

Future work building on this article might include making ITSs less expensive to build by accomplishing more development tasks with machine learning, or making POMDPs useful for even more tutoring problems by addressing the performance degradation state compression information loss can cause. Immediate next steps will include building a POMDP to drive a real-world ITS for the CFF instructional domain, and evaluating its efficacy with human trainees.

REFERENCES

ALMOND, R. G., SHUTE, V. J., UNDERWOOD, J. S., AND ZAPATA-RIVERA, J.-D. 2009. Bayesian networks: A teacher's view. *Int. J. Approx. Reason. 50*, 450–460.

ANDERSON, J. R. 1993. *Rules of the Mind*. Lawrence Erlbaum Associates, Hillsdale, NJ.

ARROYO, I., BEAL, C. R., MURRAY, T., WALLES, R., AND WOOLF, B. P. 2004. Wayang outpost: Intelligent tutoring for high stakes achievement tests. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, J. C. Lester, R. M. Vicari, and F. Paraguaçu, Eds., Springer, Berlin, 142–169.

ARROYO, I., WOOLF, B. P., AND BEAL, C. R. 2006. Addressing cognitive differences and gender during problem solving. *Technol. Instruct. Cogni. Learn. 4*, 31–63.

BAILEY, M. P. AND ARMSTRONG, R. 2002. The deployable virtual training environment. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference*.

BAKER, R. S. J. D., CORBETT, A. T., GOWDA, S. M., WAGNER, A. Z., MACLAREN, B. A., KAUFFMAN, L. R., MITCHELL, A. P., AND GIGUERE, S. 2010. Contextual slip and prediction of student performance after use of an intelligent tutor. In *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization*, P. de Bra, A. Kobsa, and D. Chin, Eds., Springer, Berlin, 52–63.

BAKER, R. S. J. D., RODRIGO, M. M. T., AND XOLOCOTZIN, U. E. 2007. The dynamics of affective transitions in simulation problem-solving environments. In *Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction*, Lisbon, A. Paiva, R. Prada, and R. W. Picard, Eds., Springer, 666–677.

BOUTILIER, C., DEAN, T., AND HANKS, S. 1999. Decision-Theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res. 11*, 94.

BOUTILIER, C. AND POOLE, D. 1996. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the 13th National Conference on Artificial Intelligence*, AAAI Press/MIT Press, 1168–1175.

CASTELLANO, M., MASTRONARDI, G., DI GIUSEPPE, G., AND DICENSI, V. 2007. Neural techniques to improve the formative evaluation procedure in intelligent tutoring systems. In *Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. 63–67.

CHA, H. J., KIM, Y. S., PARK, S. H., YOON, T. B., JUNG, Y. M., AND LEE, J. 2006. Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds., Springer, Berlin, 513–524.

CHI, M. T. H., FELTOVICH, P. J., AND GLASER, R. 1981. Categorization and representation of physics problems by experts and novices. *Cogn. Sci. 5*, 121–152.

Conati, C. 2002. Probabilistic assessment of user's emotions in educational games. *App. Artifi. Intell. 16*, 555–575.

Conati, C., Gertner, A., and Vanlehn, K. 2002. Using bayesian networks to manage uncertainty in student modeling. *User Model. User-Adapt. Interact. 12*, 371–417.

Conati, C. and Maclaren, H. 2009. Empirically building and evaluating a probabilistic model of user affect. *User Model. User-Adapt. Interact. 19*, 267–303.

Corbett, A. T. and Anderson, J. R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact. 4*, 253–278.

Craig, S. D., Graesser, A. C., Sullins, J., and Gholson, B. 2004. Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *J. Educ. Media 29*, 241–250.

D'Mello, S. K., Craig, S. D., Sullins, J., and Graesser, A. C. 2006. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *Int. J. Artif. Intell. Educ. 16*, 3–28.

D'Mello, S. K., Taylor, R. S., and Graesser, A. C. 2007. Monitoring affective trajectories during complex learning. In *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, D. S. McNamara and J. G. Trafton, Eds., Cognitive Science Society, 203–208.

Ericsson, K. A. and Simon, H. A. 1984. *Protocol Analysis*. MIT Press, Cambridge, MA.

Feng, Z. and Hansen, E. 2004. An approach to state aggregation for pomdps. In *Proceedings of the AAAI-04 Workshop on Learning and Planning in Markov Processes–Advances and Challenges*, D. P. De Farias, S. Mannor, D. Precup, and G. Theocharous, Eds., AAAI Press.

Ferguson, K., Arroyo, I., Mahadevan, S., Woolf, B. P., and Barto, A. 2006. Improving intelligent tutoring systems: Using expectation maximization to learn student skill levels. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds., Springer, 453–462.

Fonteyn, M. E., Kuipers, B., and Grobe, S. J. 1993. A description of think aloud method and protocol analysis. *Qual. Health Res. 3*, 430–441.

Graesser, A. C., Mcdaniel, B., Chipman, P., Witherspoon, A., D'Mello, S. K., and Gholson, B. 2006. Detection of emotions during learning with AutoTutor. In *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, R. Son, Ed., Erlbaum, Mahwah, NJ, 285–290.

Graesser, A. C. and Person, N. K. 1994. Question asking during tutoring. *Amer. Educ. Res. J. 31*, 104–137.

Hattie, J. and Timperley, H. 2007. The power of feedback. *Rev. Educ. Res. 77*, 81–112.

Hatzilygeroudis, I. and Prentzas, J. 2004. Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. *Expert Syst. Appl. 26*, 477–492.

Hauskrecht, M. and Fraser, H. 2000. Planning treatment of ischemic heart disease with partially observable Markov decision processes. *Artif. Intell. Med. 18*, 221–244.

Hauskrecht, M., Meuleau, N., Boutilier, C., Kaelbling, L. P., and Dean, T. 1998. Hierarchical solution of markov decision processes using macro-actions. In *Proceedings of the 14th International Conference on Uncertainty In Artificial Intelligence*, G. Cooper and S. Moral, Eds., Morgan Kaufmann Publishers, San Francisco, CA.

He, R., Brunskill, E., and Roy, N. 2010. PUMA: Planning under uncertainty with macro-actions. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAI Press.

Hoey, J. and Little, J. J. 2007. Value-directed human behavior analysis from video using partially observable markov decision processes. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 1118–1132.

Hoey, J. and Poupart, P. 2005. Solving POMDPs with continuous or large discrete observation spaces. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, L. P. Kaelbling and A. Saffiotti, Eds., Professional Book Center.

Johnson, W. L. 1990. Understanding and debugging novice programs. *Artif. Intell. 42*, 51–97.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artif. Intell. 101*, 99–134.

Kurniawati, H., Hsu, D., and Lee, W. S. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of the 4th Robotics: Science and Systems Conference*, O. Brock, J. Trinkle, and F. Ramos, Eds., MIT Press, 65–72.

Lee, C. S. 2007. Diagnostic, predictive and compositional modeling with data mining in integrated learning environments. *Comput. Educ. 49*, 562–580.

Levchuk, G., Shebilske, W., and Freeman, J. 2013. A model-driven instructional strategy: The benchmarked experiential system for training (BEST). In *Adaptive Technologies for Training and Education*, P. J. Durlach and A. M. Lesgold, Eds., Cambridge University Press, Cambridge, UK.

LOVEJOY, W. S. 1991. Computationally feasible bounds for partially observed Markov decision processes. *Oper. Res. 39*, 162–175.

LUCKIN, R. AND DU BOULAY, B. 1999. Ecolab: The development and evaluation of a Vygotskian design framework. *Int. J. Artif. Intell. Educ. 10*, 198–220.

MAYO, M. AND MITROVIC, A. 2001. Optimising ITS behaviour with Bayesian networks and decision theory. *Int. J. Artif. Intell. Educ. 12*, 124–153.

MCCALLUM, R. A. 2002. Hidden state and reinforcement learning with instance-based state identification. *IEEE Trans. Syst. Man. Cybernet. 26*, 464–473.

MCQUIGGAN, S. W., MOTT, B. W., AND LESTER, J. C. 2008. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Model. User-Adapt. Interact. 18*, 81–123.

NORMAN, D. A. 1981. Categorization of action slips. *Psychol. Rev. 88*, 1–15.

PAAS, F. G. W. C. 1992. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *J. Educ. Psychol. 84*, 429–434.

PAYNE, S. J. AND SQUIBB, H. R. 1990. Algebra mal-rules and cognitive accounts of error. *Cogn. Sci. 14*, 445–481.

PINEAU, J., GORDON, G., AND THRUN, S. 2003. Point-Based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1025–1032.

POUPART, P. 2005. Exploiting structure to efficiently solve large scale partially observable markov decision processes. Unpublished doctoral dissertation. Department of Computer Science, University of Toronto, Toronto, Canada. 156 pages.

RITTER, S., HARRIS, T. K., NIXON, T., DICKISON, D., MURRAY, R. C., AND TOWLE, B. 2009. Reducing the knowledge tracing space. In *Proceedings of the 2nd International Conference on Educational Data Mining*, T. Barnes, M. Desmarais, C. Romero, and S. Ventura, Eds., 151–160.

ROBISON, J., MCQUIGGAN, S., AND LESTER, J. 2009. Evaluating the consequences of affective feedback in intelligent tutoring systems. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction*. IEEE.

RODRIGO, M. M. T., BAKER, R. S. J. D., LAGUD, M. C. V., AL LIM, S., MACAPANPAN, A. F., PASCUA, S., SANTILLANO, J. Q., SEVILLA, L. R. S., SUGAY, J. O., AND SINATH, T. E. P. 2007. Affect and usage choices in simulation problem-solving environments. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, R. Luckin, K. R. Koedinger, and J. E. Greer, Eds., IOS Press, 145–152.

SHEBILSKE, W., GILDEA, K., FREEMAN, J., AND LEVCHUK, G. 2009. Optimising instructional strategies: A benchmarked experiential system for training. *Theor. Issue. Ergon. Sci. 10*, 267–278.

SHUTE, V. J. AND GLASER, R. 1990. A large-scale evaluation of an intelligent discovery world: Smithtown. *Interact. Learn. Environ. 1*, 51–77.

SHUTE, V. J., HANSEN, E. G., AND ALMOND, R. G. 2008. You can't fatten a hog by weighing it – Or can you? evaluating an assessment for learning system called ACED. *Int. J. Artif. Intell. Educ. 18*, 289–316.

SNOW, R. E. AND LOHMAN, D. F. 1984. Toward a theory of cognitive aptitude for learning from instruction. *J. Educ. Psychol. 76*, 347–376.

THEOCHAROUS, G., BECKWITH, R., BUTKO, N., AND PHILIPOSE, M. 2009. Tractable POMDP planning algorithms for optimal teaching in "SPAIS." In *Proceedings of the 21st International Joint Conferences on Artificial Intelligence Workshop on Plan, Activity, and Intent Recognition*, C. Boutilier, Ed., AAAI Press, Menlo Park, CA.

THEOCHAROUS, G., BUTKO, N., AND PHILIPOSE, M. 2010. Designing a mathematical manipulatives tutoring system using POMDPs. In *Proceedings of the POMDP Practitioners Workshop on Solving Real-world POMDP Problems at the 20th International Conference on Automated Planning and Scheduling*, R. Brafman, H. Geffner, J. Hoffmann, and H. Kautz, Eds., AAAI Press, Menlo Park, CA.

THRUN, S., BEETZ, M., BENNEWITZ, M., BURGARD, W., CREMERS, A. B., DELLAERT, F., FOX, D., HÄHNEL, D., ROSENBERG, C., AND ROY, N. 2000. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *Int. J. Robot. Res. 19*, 972–1007.

U. S. DEPARTMENT OF EDUCATION 2008. What works clearinghouse: Procedures and standards handbook (Version 2.0). http://ies.ed.gov/ncee/wwc/references/idocviewer/doc.aspx?docid=19&tocid=1 (last accessed 8/09).

VANLEHN, K. AND MARTIN, J. 1998. Evaluation of an assessment system based on Bayesian student modeling. *Int. J. Artif. Intell. Educ. 8*, 179–221.

VANLEHN, K. AND NIU, Z. 2001. Bayesian student modeling, user interfaces and feedback: A sensitivity analysis. *Int. J. Artif. Intell. Educ. 12*, 154–184.

VOGEL-WALCUTT, J. J., FIORE, S., BOWERS, C., AND NICHOLSON, D. 2009. Embedding metacognitive prompts during SBT to improve knowledge acquisition. In *Proceedings of the 53rd Annual Meeting of the Human Factors and Ergonomics Society*. Human Factors and Ergonomics Society, 1939–1943.

VOGEL-WALCUTT, J. J., SCHATZ, S., BOWERS, C., GEBRIM, J. B., SCIARINI, L. W., AND NICHOLSON, D. 2008. Augmented cognition and training in the laboratory: DVTE system validation. In *Proceedings of the 52nd Annual Meeting of the Human Factors and Ergonomics Society*. Human Factors and Ergonomics Society, 187–191.

WEI, L. Z. 2010. Download - APPL. http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/index.php?n=Main.Download

YOUNG, S., GASIC, M., KEIZER, S., MAIRESSE, F., SCHATZMANN, J., THOMSON, B., AND YU, K. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Comput. Speech Lang. 24*, 150–174.

ZAKHAROV, K., MITROVIC, A., AND JOHNSTON, L. 2008. Towards emotionally-intelligent pedagogical agents. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, Eds., Springer, 19–28.

Received September 2010; revised May 2011; accepted January 2012