Symbolic Dynamic Programming for Continuous State and Observation POMDPs

Authored by:

Pascal Poupart
Scott Sanner
Zahra Zamani
Kristian Kersting

Abstract

Partially-observable Markov decision processes (POMDPs) provide a powerful model for real-world sequential decision-making problems. In recent years, point- based value iteration methods have proven to be extremely effective techniques for ?nding (approximately) optimal dynamic programming solutions to POMDPs when an initial set of belief states is known. However, no point-based work has provided exact point-based backups for both continuous state and observation spaces, which we tackle in this paper. Our key insight is that while there may be an in?nite number of possible observations, there are only a ?nite number of observation partitionings that are relevant for optimal decision-making when a ?nite, ?xed set of reachable belief states is known. To this end, we make two important contributions: (1) we show how previous exact symbolic dynamic pro- gramming solutions for continuous state MDPs can be generalized to continu- ous state POMDPs with discrete observations, and (2) we show how this solution can be further extended via recently developed symbolic methods to continuous state and observations to derive the minimal relevant observation partitioning for potentially correlated, multivariate observation spaces. We demonstrate proof-of- concept results on uni- and multi-variate state and observation steam plant control.

1 Paper Body

Point-based value iteration (PBVI) methods have proven extremely effective for finding (approximately) optimal dynamic programming solutions to partiallyobservable Markov decision processes (POMDPs) when a set of initial belief states is known. However, no PBVI work has provided exact point-based backups for both continuous state and observation spaces, which we tackle in this paper. Our key insight is that while there may be an infinite number of observations, there are only a finite number of continuous observation partitionings that are

relevant for optimal decision-making when a finite, fixed set of reachable belief states is considered. To this end, we make two important contributions: (1) we show how previous exact symbolic dynamic programming solutions for continuous state MDPs can be generalized to continuous state POMDPs with discrete observations, and (2) we show how recently developed symbolic integration methods allow this solution to be extended to PBVI for continuous state and observation POMDPs with potentially correlated, multivariate continuous observation spaces.

1

Introduction

Partially-observable Markov decision processes (POMDPs) are a powerful modeling formalism for real-world sequential decision-making problems [3]. In recent years, point-based value iteration methods (PBVI) [5, 10, 11, 7] have proved extremely successful at scaling (approximately) optimal POMDP solutions to large state spaces when a set of initial belief states is known. While PBVI has been extended to both continuous state and continuous observation spaces, no prior work has tackled both jointly without sampling. [6] provides exact point-based backups for continuous state and discrete observation problems (with approximate sample-based extensions to continuous actions and observations), while [2] provides exact point-based backups (PBBs) for discrete state and continuous observation problems (where multivariate observations must be conditionally independent). While restricted to discrete states, [2] provides an important insight that we exploit in this work: only a finite number of partitionings of the observation space are required to distinguish between the optimal conditional policy over a finite set of belief states. We propose two major contributions: First, we extend symbolic dynamic programming for continuous state MDPs [9] to POMDPs with discrete observations, arbitrary continuous reward and transitions with discrete noise (i.e., a finite mixture of deterministic transitions). Second, we extend this symbolic dynamic programming algorithm to PBVI and the case of continuous observations 1

(while restricting transition dynamics to be piecewise linear with discrete noise, rewards to be piecewise constant, and observation probabilities and beliefs to be uniform) by building on [2] to derive relevant observation partitions for potentially correlated, multivariate continuous observations.

2

Hybrid POMDP Model

A hybrid (discrete and continuous) partially observable MDP (H-POMDP) is a tuple hS, A, O, T , R, Z, ?, hi. States S are given by vector (ds , xs) = (ds1 , . . . , dsn , xs1 , . . . , xsm) where each dsi ? {0, 1} (1 ? i ? n) is boolean and each xsj ? R (1 ? j ? m) is continuous. We assume a finite, discrete action space A = {a1 , . . . , ar }. Observations O are given by the vector (do , xo) = (do1 , . . . , dop , xo1 , . . . , xoq) where each doi ? {0, 1} (1 ? i ? p) is boolean and each xoj ? R (1 ? j ? q) is continuous. Three functions are required for modeling H-POMDPs: (1) T : S ? A ? S ? [0, 1] a Markovian transition model defined as the probability of the next state given the action and previous state; (2) R : S ? A ? R a reward function which returns the immediate reward of

taking an action in some state; and (3) an observation function defined as Z : S ? A ? O ? [0, 1] which gives the probability of an observation given the outcome of a state after executing an action. A discount factor ?, 0 ? ? ? 1 is used to discount rewards t time steps into the future by ? t . We use a dynamic Bayes net (DBN)1 to compactly represent the transition model T over the factored state variables and we use a two-layer Bayes net to represent the observation model Z: T : p(x0s ,d0s —xs ,ds , a) =

n Y

p(d0si —xs ,ds , a)

i=1

Z : p(xo ,do —x0s ,d0s , a) =

p Y

m Y

p(x0sj —xs ,ds , d0s , a).

(1)

p(xoj —x0s ,d0s , a).

(2)

j=1

p(doi —x0s ,d0s , a)

i=1

q Y j=1

Probabilities over discrete variables p(d0si—xs ,ds ,a) and p(doi—x0s ,d0s ,a) may condition on both discrete variables and (nonlinear) inequalities of continuous variables; this is further restricted to linear inequalities in the case of continuous observations. Transitions over continuous variables p(x0sj—xs ,ds ,d0s ,a) must be deterministic (but arbitrary nonlinear) piecewise functions; in the case of continuous observations they are further restricted to be piecewise linear; this permits discrete noise in the continuous transitions since they may condition on stochastically sampled discrete next-state variables d0s . Observation probabilities over continuous variables p(xoj—x0s ,d0s ,a) only occur in the case of continuous observations and are required to be piecewise constant (a mixture of uniform distributions); the same restriction holds for belief state representations. The reward R(d, x, a) may be an arbitrary (nonlinear) piecewise function in the case of deterministic observations and a piecewise constant function in the case of continuous observations. We now provide concrete examples. Example (Power Plant) [1] The steam generation system of a power plant evaporates feed-water under restricted pressure and temperature conditions to turn a steam turbine. A reward is obtained when electricity is generated from the turbine and the steam pressure and temperature are within safe ranges. Mixing water and steam makes the respective pressure and temperature observations po ? R and to ? R on the underlying state ps ? R and ts ? R highly uncertain. Actions A = {open, close} control temperature and pressure by means of a pressure valve. We initially present two H-POMDP variants labeled 1D-Power Plant using a single temperature state variable ts . The transition and reward are common to both ? temperature increments (decrements) with a closed (opened) valve, a large negative reward is given for a closed valve

with ts exceeding critical threshold 15, and positive reward is given for a safe, electricity-producing state: ” p(t0s —ts , a) = ? t0s ?

(

(a = open) : ts ? 5 (a = close) : ts + 7

#

? ? : ?1 ?(a = open) R(ts , a) = (a = close) ? (ts ¿ 15) : ?1000 (3) ? ?(a = close) ? ?(t ¿ 15) : 100 s

Next we introduce the Discrete Obs. 1D-Power Plant variant where we define an observation space with a single discrete binary variable o ? O = {high, low }: 1 We disallow general synchronic arcs for simplicity of exposition but note their inclusion only places restrictions on the variable elimination ordering used during the dynamic programming backup operation.

2

b

(1 * x) ¿= 50

(1 * x) ¡= 39

234 + (1.5 * x)

(1 * x) ¿= 150

250

(1 * x) ¡= 139

197 + (2 * x)

121 + (3 * x)

Figure 1: (left) Example conditional plan ? h for discrete observations; (right) example ?-function for ? h over state b ? {0, 1}, x ? R in decision diagram form: the true (1) branch is solid, the false (0) branch is dashed. (p(o = high—t0s , a = open) =

t0s ? 15 : 0.9 t0s ¿ 15 : 0.1

(p(o = high—t0s , a = close) =

t0s ? 15 : 0.7 t0s ¿ 15 : 0.3

(4)

Finally we introduce the Cont. Obs. 1D-Power Plant variant where we define an observation space with a single continuous variable to uniformly distributed on an interval of 10 units centered at t0s . (p(to —t0s , a

= open) =

U (to ; t0s

?

5, t0s

+ 5) =

(to ¿ t0s ? 5) ? (to ¡ t0s + 5) : 0.1 (to ? t0s ? 5) ? (to ? t0s + 5) : 0

(5)

While simple, we note no prior method could perform exact point-based backups for either problem.

3

Value Iteration for Hybrid POMDPs

In an H-POMDP, the agent does not directly observe the states and thus must maintain a belief state b(xs ,ds) = p(xs ,ds). For a given belief state b

= b(xs ,ds), a POMDP policy ? can be represented by a tree corresponding to a conditional plan ?. An h-step conditional plan ? h can be defined recursively in terms of (h ? 1)-step conditional plans as shown in Fig. 1 (left). Our goal is to find a policy ? that maximizes the value function, defined as the sum of expected discounted rewards over horizon h starting from initial belief state b: V?h (b) = E?

hXh t=0

i

? t ? rt b0 = b

(6)

where rt is the reward obtained at time t and b0 is the belief state at t = 0. For finite h and belief state b, the optimal policy ? is given by an h-step conditional plan ? h . For h = ?, the optimal discounted (? ¡ 1) value can be approximated arbitrarily closely by a sufficiently large h [3]. Even when the state is continuous (but the actions and observations are discrete), the optimal POMDP value function for finite horizon h is a piecewise linear and convex function of the belief state b [6], hence V h is given by a maximum over a finite set of ??-functions? ?ih : h

V (b) = max

h ?h i ??

h?ih , bi

Z = max

h ?h i ??

X

?ih (xs ,ds) ? b(xs ,ds) dxs

(7)

xs d s

Later on when we tackle continuous state and observations, we note that we will dynamically derive an optimal, finite partitioning of the observation space for a given belief state and hence reduce the continuous observation problem back to a discrete observation problem at every horizon. The ?h in this optimal h-stage-to-go value function can be computed via Monahan?s dynamic programming approach to value iteration (VI) [4]. Initializing ?10 = 0, ?0 = {?10 }, and assuming discrete observations o ? Oh , ?h is obtained from ?h?1 as follows:2 h ga,o,j (xs ,ds) =

Z

X

p(o—x0s ,d0s , a)p(x0s ,d0s —xs ,ds , a)?jh?1 (x0s ,d0s)dxs0 ; ??jh?1 ? ?h?1

(8)

xs0 d s0

n o h ?ha = R(xs ,ds , a) + ? o?O ga,o,j (xs ,ds) j [h ?h = ?a

(9) (10)

a 2 The of sets is defined as {p + q—p ? P, q ? Q}.

j?{1,...,n} Sj

= S1 ? ? ? Sn where the pairwise cross-sum P Q =

3

Algorithm 1: PBVI(H-POMDP, H, B = {bi }) ?? hV h i 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

begin V 0 := 0, h := 0, ?0P BV I = {?10 } while h ¡ H do h := h + 1, ?h := ?, ?hP BV I := ? foreach bi ? B do foreach a ? A do ?ha := ? if (continuous observations: q ¿ 0) then // Derive relevant observation partitions Oih for belief bi hOih , p(Oih —x0s ,d0s , a)i := GenRelObs(?h?1 P BV I , a, bi) else // Discrete observations and model already known Oih := {do }; p(Oih —x0s ,d0s , a) := see Eq (2) foreach o ? Oih do foreach ?jh?1 ? ?h?1 P BV I do ?jh?1 := Prime(?jh?1) // ?di : di ? d0i and ?xi : xi ? x0i h ga,o,j := see Eq (8)

18

?ha := see Eq (9) ?h := ?h ? ?ha

19 20 21

// Retain only ?-functions optimal at each belief point foreach bi ? B do h ?b := arg max?j ??h ?j ? bi i h h ?P BV I := ?hP BV I ? ?b i

22 23 24 25 26 27

// Terminate if early convergence if ?hP BV I = ?h?1 P BV I then break

28 29 30

return ?P BV I

31 32

end

Point-based value iteration (PBVI) [5, 11] computes the value function only for a set of belief states {bi } where bi := p(xs ,ds). The idea is straightforward and the main modification needed to Monahan?s VI approach in Algorithm 1 is the loop from lines 23?25 where only ?-functions optimal at some belief state are retained for subsequent iterations. In the case of continuous observation variables (q ¿ 0), we will need to derive a relevant set of observations on line 10, a key contribution of this work as described in Section 4.3. Otherwise if the observations are only discrete (q = 0), then a finite set of observations is already known and the observation function as given in Eq (2). We remark that Algorithm 1 is a generic framework that can be used for both PBVI and other variants of approximate VI. If used for PBVI, an efficient direct backup computation of the optimal ?-function for belief state bi should be used in line 17 that is linear in the number of observations [5, 11] and which obviates the need for lines 23?25. However, for an alternate version of approximate value iteration that will often produce more accurate values for belief states other than those in B, one may instead retain the full cross-sum backup of line 17, but omit lines 23?25 ? this yields an approximate VI approach (using discretized observations relevant only to a chosen set of belief states B if continuous observations are present) that is not restricted to alpha-functions only optimal at B, hence allowing greater flexibility in approximating the value function over all belief states. Whereas PBVI is optimal if all reachable belief states within horizon H are enumerated in B, in the H-POMDP setting, the generation of continuous observations will most often lead to an infinite number of reachable belief states, even with finite horizon ? this makes it quite difficult to provide optimality guarantees in the general case of PBVI for continuous observation

settings. Nonetheless, PBVI has been quite successful in practice without exhaustive enumeration of all reachable beliefs [5, 10, 11, 7], which motivates our use of PBVI in this work. 4

4

Symbolic Dynamic Programming

In this section we take a symbolic dynamic programming (SDP) approach to implementing VI and PBVI as defined in the last section. To do this, we need only show that all required operations can be computed efficiently and in closed-form, which we do next, building on SDP for MDPs [9]. 4.1

Case Representation and Extended ADDs

The previous Power Plant examples represented all functions in case form, generally defined as ? ? ? : ? ? 1 . f = .. ? ? ?? : k

f1 .. . fk

and this is the form we use to represent all functions in an H-POMDP. The ?i are disjoint logical formulae defined over xs ,ds and/or xo ,do with logical (?, ?, ?) combinations of boolean variables and inequalities (?, ¿, ?, ¡) over continuous variables. For discrete observation H-POMDPs, the fi and inequalities may use any function (e.g., sin(x1) ¿ log(x2) ? x3); for continuous observations, they are restricted to linear inequalities and linear or piecewise constant fi as described in Section 2. For unary operations such as scalar multiplication c ? f (for some constant c ? R) or negation ?f on case statements is simply to apply the operation on each case partition fi (1 ? i ? k). A binary operation on two case statements, takes the cross-product of the logical partitions of each case statement and performs the corresponding operation on the resulting paired partitions. The cross-sum ? of two cases is defined as the following: ? ((?1 : f 1 ?1 : g 1 ? ?2 : f 2 ?2 : g 2

? ? 1 ? ?1 ? ? ?? ? ? 1 2 = ? ? 2 ? ?1 ? ? ?? ? ? 2 2

: : : :

f1 + g1 f1 + g2 f2 + g1 f2 + g2

Likewise and ? are defined by subtracting or multiplying partition values. Inconsistent partitions can be discarded when they are irrelevant to the function ? value. A symbolic case maximization is defined as below: ??1 ? ?1 ? f1 ¿ g1 : f1 (casemax

?1 : f1 , ?2 : f2

(

?1 : g1 ?2 : g2

!

? ? ? ? ? ? ?1 ? f1 ? g1 : g1 ? ? 1 = ? 1 ? ?2 ? f 1 ¿ g 2 : f 1 ? ? ?1 ? ?2 ? f1 ? g2 : g2 ? ? ? ? .. ?.. . .

The following SDP operations on case statements require more detail than can be provided here, hence we refer the reader to the relevant literature: ? Substitution f ?: Takes a set ? of variables and their substitutions (which may be case statements themselves), and carries out all variable substitutions [9]. R ? Integration x1 f dx1 : There are two forms: If x1 is involved in a ?-function (cf. the transition in Eq (3)) then the integral is equivalent to a symbolic substitution and can be applied to any case statement (cf. [9]). Otherwise, if f

is in linearly constrained polynomial case form, then the approach of [8] can be applied to yield a result in the same form. Case operations yield a combinatorial explosion in size if na??vely implemented, hence we use the data structure of the extended algebraic decision diagram (XADD) [9] as shown in Figure 1 (right) to compactly represent case statements and efficiently support the above case operations with them. 4.2

VI for Hybrid State and Discrete Observations

For H-POMDPs with only discrete observations o ? O and observation function p(o—x0s ,d0s , a) as in the form of Eq (4), we introduce a symbolic version of Monahan?s VI algorithm. In brief, we note that all VI operations needed in Section 3 apply directly to H-POMDPs, e.g., rewriting Eq (8): Z

h ga,o,j (xs ,ds) =

" M xs0 d

s0

p(o—x0s ,d0s ,a)?

n O

! p(d0si—xs ,ds ,a)

i=1

?

m O

! p(x0sj—xs ,ds , d0s ,a)

#

??jh?1 (x0s ,d0s)

dxs0

j=1

(11)

5

Algorithm 2: GenRelObs(?h?1 , a, bi) ?? hOh , p(Oh —x0s ,d0s , a)i 1 2 3 4 5 6 7 8 9 10 11 12 13 14

begin foreach ?j (x0s ,d0s) ? ?h?1 and a ? A do // Perform exact 1-stepRDPL backup of ?-functions at horizon h ? 1 0 0 0 0 0 0 0 ?ja (xs ,ds , xo ,do) := x0 d0s p(xo ,do —xs ,ds , a) ? p(xs ,ds —xs ,ds , a) ? ?j (xs ,ds) dxs s a foreach ?j (xs ,ds , xo ,do) do // Generate value each ?-vector at belief point bi (xs ,ds) as a function of observations R ofL ?ja (xo ,do) := xs ds bi (xs ,ds) ? ?ja (xs ,ds , xo ,do) dxs // Using casemax, generate observation partitions relevant to each policy ? see text for details Oh := extract-partition-constraints[casemax(?1a1 (xo ,do), ?1a2 (xo ,do), . . . , ?jar (xo ,do))] foreach ok ? Oh do // Let ?ok be the partition constraints for observation ok ? Oh R L p(Oh = ok —x0s ,d0s , a) := xo do p(xo ,do —x0s ,d0s , a)I[?ok]dxo return hOh , p(Oh —x0s ,d0s , a)i end

P(t s)

(t o) P(o) =0.0127

b1

0.25 0.2

b

7.5 0.1 0

2
2
P(o) =0.983 1
close
-72.5 -75
2
6
ts
open
4
11
5 5.1
8
to
14 15
18

Figure 2: (left) Beliefs b1 , b2 for Cont. 1D-Power Plant; (right) derived observation partitions for b2 at h = 2.

Crucially we note since the continuous transition cpfs p(x0sj—xs ,ds ,d0s ,a) are deterministic and hence R defined with Dirac ??s (e.g., Eq 3) as described in Section 2, the integral x 0 can always be computed s in closed case form as discussed in Section 4.1. In short, nothing additional is required for PBVI on H-POMDPs in this case ? the key insight is simply that ?-functions are now represented by case statements and can ?grow? with the horizon as they partition the state space more and more finely. 4.3

PBVI for Hybrid State and Hybrid Observations

In general, it would be impossible to apply standard VI to H-POMDPs with continuous observations since the number of observations is infinite. However, building on ideas in [2], in the case of PBVI, it is possible to derive a finite set of continuous observation partitions that permit exact point-based backups at a belief point. This additional operation (GenRelObs) appears on line 10 of PBVI in Algorithm 1 in the case of continuous observations and is formally defined in Algorithm 2. To demonstrate the generation of relevant continuous observation partitions, we use the second iteration of the Cont. Obs. 1D-Power Plant along with two belief points represented as uniform distributions: b1 : U (ts ; 2, 6) and b2 : U (ts ; 6, 11) as shown in Figure 2 (left). Letting h = 2, we will assume simply for expository purposes that —?1 — = 1 (i.e., it contains only one ?-function) and that in lines 2?4 of Algorithm 2 we have computed the following two ?-functions for a ? {open, close}: ? (? ?(ts ¡ 15) ? (ts ?10 ¡ to ¡ ts) : 10 (ts ?10 ¡ to ¡ ts) : 0.1 open close ?1 (ts , to) = (ts ? 15) ? (ts ?10 ¡ to ¡ ts) : ?100 ?1 (ts , to) = ? ?(ts ?10 ¡ to ¡ ts) : 0 ??(t ?10 ¡ t ¡ t) :0 s o s

We now need the ?-vectors as a function of the observation space for a particular belief state, thus next we marginalize out xs ,ds in lines 5?7. The resulting ?-functions are shown as follows where for brevity from this point forward, 0 partitions are suppressed in the cases: 6

? ? ?(14 ¡ to ¡ 18) ?1close (to) = (8 ¡ to ¡ 14) ? ?(4 ¡ t ¡ 8) o

? ? ?(15 ¡ to ¡ 18) ? ? ? ?(14 ¡ to ¡ 15) open ?1 (to) = (8 ¡ to ¡ 14) ? ? ?(5 ¡ to ¡ 8) ? ? ? (4 ¡ to ¡ 5)

: 0.025to ? 0.45 : ?0.1 : ?0.025to ? 0.1

: 25to ? 450 : ?2.5to ? 37.5 : ?72.5 : ?25to + 127.5 : 2.5to ? 10

Both ?1close (to) and ?1open (to) are drawn graphically in Figure 2 (right). These observationdependent ??s divide the observation space into regions which can yield the optimal policy according to the belief state b2 . Following [2], we need to find the optimal boundaries or partitions of the observation space; in their work, numerical solutions are proposed to find these boundaries in one dimension (multiple observations are handled through an independence assumption). Instead, here we leverage the symbolic power of the casemax operator defined in Section 4.1 to find all the partitions where each potentially correlated, multivariate observation ? is optimal. For the two ??s above, the following partitions of the observation space are derived by the casemax operator in line 9: ? ? o1 ? ? ?o ? 1

? casemax ?1close (to), ?1open (to) = o1 ? ? ? o2 ? ? ? o2

: (14 ¡ to ? 18) : 0.025to ? 0.45 : (8 ¡ to ? 14) : ?0.1 : (5.1 ¡ to ? 8) : ?0.025to ? 0.1 : (5 ¡ to ? 5.1) : ?25to + 127.5 : (4 ¡ to ? 5) : 2.5to ? 10

Here we have labeled with o1 the observations where ?1close is maximal and with o2 the observations where ?1open is maximal. What we really care about though are just the constraints identifying o1 and o2 and this is the task of extract-partition-constraints in line 9. This would associate with o1 the partition constraint ?o1 ? (5.1 ¡ to ? 8) ? (8 ¡ to ? 14) ? (14 ¡ to ? 18) and with o2 the partition constraint ?o2 ? (4 ¡ to ? 5) ? (5 ¡ to ? 5.1) ? taking into account the 0 partitions and the 1D nature of this example, we can further simplify ?o1 ? (to ¿ 5.1) and ?o2 ? (to ? 5.1). Given these relevant observation partitons, our final task in lines 10-12 is to compute the probabilities of each observation partition ?ok . This is simply done by marginalizing over the observation function p(Oh —x0s ,d0s , a) within each region defined by ?ok (achieved by multiplying by an indicator function I[?ok] over these constraints). To better understand what is computed here, we can compute the probability p(ok —bi , a) of each observation for a particular belief, calculated as follows: Z

Z

MM

p(ok —bi , a) := xs

x0s d s

p(ok —x0s ,d0s , a)?p(x0s ,d0s —xs ,ds , a)??j (x0s ,d0s)?bi (xs ,ds) dx0s dxs (12)

d0s

Specifically, for b2 , we obtain p(o1 —b2 , a = close) = 0.0127 and p(o2 —b2 , a = close) = 0.933 as shown in Figure 2 (right). In summary, in this section we have shown how we can extend the exact dynamic programming algorithm for the continuous state, discrete observation POMDP setting from Section 4.2 to compute exact 1-step point-based backups in the continuous observation setting; this was accomplished through the crucial insight that despite the infinite

number of observations, using Algorithm 2 we can symbolically derive a set of relevant observations for each belief point that distinguish the optimal policy and hence value as graphically illustrated in Figure 2 (right). Next we present some empirical results for 1- and 2-dimensional continuous state and observation spaces.

5

Empirical Results

We evaluated our continuous POMDP solution using XADDs on the 1D-Power Plant example and another variant of this problem with two variables, described below.3 2D-Power Plant: We consider the more complex model of the power plant similar to [1] where the pressure inside the water tank must be controlled to avoid mixing water into the steam (leading to explosion of the tank). We model an observable pressure reading po as a function of the underlying pressure state ps . Again we have two actions for opening and closing a pressure valve. The close action has transition " p(p0s —ps , a

= close) = ?

(p0s

?

(p + 10 ¿ 20) : 20 ?(p + 10 ¿ 20) : ps + 10

#

p(t0s —ts , a = close) = ? t0s ? (ts + 10)

3 Full problem specifications and Java code to reproduce these experiments are available online in Google Code: http://code.google.com/p/cpomdp .

7

Power Plant

5

Power Plant

10

Number of Nodes

Time(ms)

1 state & 1 observ var 2 state & 2 observ vars 4

10

3

10

2

10

1

2

3

4

5

6

70

1 state & 1 observ var 2 state & 2 observ vars

60

50

40
30
20
10
0 1
Horizon
2
3
4
Horizon
5
6

Figure 3: (left) time vs. horizon, and (right) space (total # XADD nodes in ?-functions) vs. horizon. and yields high reward for staying within ?the safe temperature and pressure range: ? (5 ? ps ? 15) ? (95 ? ts ? 105) : 50 ? ? ?(5 ? p ? 15) ? (t ? 95) : ?1 s s R(ts , ps , a = close) = ?(ps ? 15) : ?5 ? ? ?else : ?3

Alternately, for the open action, the transition functions reduce the temperature by 5 units and the pressure by 10 units as long as the pressure stays above zero. For the open reward function, we assume that there is always a small constant penalty (-1) since no electricity is produced. Observations are distributed uniformly within a region depending on their underlying state: p(to —t0s)

((ts + 80 ¡ to ¡ ts + 105) = ?(ts + 80 ¡ to ¡ ts + 105)

: 0.04 :0

p(po —p0s)

((ps ¡ po ¡ ps + 10) : 0.1 = ?(ps ¡ po ¡ ps + 10) : 0

Finally for PBVI, we define two uniform beliefs as follows: b1 : U [ts ; 90, 100] ? U [ps ; 0, 10] and b2 : U [ts ; 90, 130] ? U [ps ; 10, 30] In Figure 3, a time and space analysis of the two versions of Power Plant have been performed for up to horizon h = 6. This experimental evaluation relies on one additional approximation over the PBVI approach of Algorithm 1 in that it substitutes p(Oh —b, a) in place of p(Oh —x0s ,d0s , a) ? while this yields correct observation probabilities for a point-based backup at a particular belief state b, the resulting ?-functions represent an approximation for other belief states. In general, the PBVI framework in this paper does not require this approximation, although when appropriate, using it should increase computational efficiency. Figure 3 shows that the computation time required per iteration generally increases since more complex ?-functions lead to a larger number of observation partitions and thus a more expensive backup operation. While an order of magnitude more time is required to double the number of state and observation variables, one can see that the PBVI approach leads to a fairly constant amount of computation time per horizon, which indicates that long horizons should be computable for any problem for which at least one horizon can be computed in an acceptable amount of time.

6

Conclusion

We presented the first exact symbolic operations for PBVI in an expressive subset of H-POMDPs with continuous state and observations. Unlike related work that has extended to the continuous state and observation setting [6], we do not approach the problem by sampling. Rather, following [2], the key contribution of this work was to define a discrete set of observation partitions on the multivariate continuous observation space via symbolic maximization techniques and derive the related probabilities using symbolic integration. An important avenue for future work is to extend these techniques to the case of continuous state, observation, and action H-POMDPs. Acknowledgments NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the ARC through the ICT Centre of Excellence program. This work was supported by the Fraunhofer ATTRACT fellowship STREAM and by the EC, FP7-248258-First-MM.

8

2 References

[1] Mario Agueda and Pablo Ibarguengoytia. An architecture for planning in uncertain domains. In Proceedings of the ICTAI 2002 Conference, Dallas,Texas, 2002. [2] Jesse Hoey and Pascal Poupart. Solving pomdps with continuous or large discrete observation spaces. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, Scotland, 2005. [3] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101:99?134, 1998. [4] G. E. Monahan. Survey of partially observable markov decision processes: Theory, models, and algorithms. Management Science, 28(1):1?16, 1982. [5] Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun. Anytime point-based approximations for large pomdps. J. Artif. Intell. Res. (JAIR), 27:335?380, 2006. [6] J. M. Porta, N. Vlassis, M.T.J. Spaan, and P. Poupart. Point-based value iteration for continuous pomdps. Journal of Machine Learning Research, 7:195220, 2006. [7] Pascal Poupart, Kee-Eung Kim, and Dongho Kim. Closing the gap: Improved bounds on optimal pomdp solutions. In In Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS-11), 2011. [8] Scott Sanner and Ehsan Abbasnejad. Symbolic variable elimination for discrete and continuous graphical models. In In Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12), Toronto, Canada, 2012. [9] Scott Sanner, Karina Valdivia Delgado, and Leliane Nunes de Barros. Symbolic dynamic programming for discrete and continuous state mdps. In Proceedings of the 27th Conference on Uncertainty in AI (UAI-2011), Barcelona, 2011. [10] Trey Smith and Reid G. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In Proc. Int. Conf. on Uncertainty in Artificial Intelligence (UAI), 2005. [11] M. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for pomdps. Journal of Articial Intelligence Research (JAIR), page

195220, 2005.
9