Proceedings of KFIS 2001 Spring Conference, 2001. 5.

상태 조직화 강화학습을 사용한 POMDP문제 해결

Solving POMDP problem using Self-organizing state RL

이승준 · 장병탁
서울대학교 컴퓨터공학부

Seung-Joon Yi and Byoung-Tak Zhang
School of Computer Science and and Engineering, Seoul National University
E-mail : sjlee@scai.snu.ac.kr, btzhang@cse.snu.ac.kr

ABSTRACT

본 논문에서는 부분적으로 관측 가능한 환경에서 사전의 모델 정보 없이 확률적인 행동 정책을 학습하는 상태 조직화 강화 학습 모델을 제안한다. 기존의 강화학습은 환경 모델을 사전에 필요로 하고 상태 전체의 관측이 필요하기 때문에 학습 이전에 문제에 대해 알아야 한다는 제약이 있다. 또한 작은 문제에 대해서는 잘 적용되지만 상태의 수가 매우 많고 부분적으로만 관측한 경우가 많은 실제 문제에는 그대로 적용하기가 불가능하다. 이러한 두 가지 단점을 해결하기 위해 본 논문에서는 사전의 모델 정보 없이 부분적인 관측값으로부터 상태와 행동 정책을 동시에 학습해 나가는 강화 학습 모델을 제안하고, 제안된 방법을 부분적으로만 관측이 가능한 미로 탐색 문제에 적용하였다.

Keywords : Reinforcement learning, Partial Observable Markov Devision Process, State Approximation, State Self-organization

I. 서 론

강화학습은(reinforcement learning) 동적인 환경 하에서 시행착오를 거쳐 환경으로부터 주어지는 보상(reward)을 최대화하기 위한 학습 방법이다. 이러한 강화 학습은 동물 행동 심리학과 최적 제어 이론 분야에 뿌리를 두고 있으며, TD learning, Q learning[3] 등의 강화학습 방법이 여러 분야에 적용되고 있다.

이러한 기존의 강화학습은 환경을 알고 있고 각 상태들에 대한 정보를 모두 저장할 수 있다는 것을 전제로 한다. 즉 사이즈가 작고 학습자에게 상태의 정의와 같은 환경이 알려져 있는 경우에는 바로 적용이 가능하지만 환경이 알려져 있지 않은 문제에는 직접적인 적용이 불가능하고 일반적인 문제의 경우 상태 공간(state space)의 크기가 너무 커져 저장 공간상의 문제와 수렴 시간이 매우 길어지는 문제가 있다. 또한 학습자가 환경을 완전히 관측할 수 없는 상황(partially observable) 에는 서로 다른 상태를 구별할 수 없는 경우(perceptual aliasing) 이 발생하여 안정된 결과를 얻기 힘들다. 본 논문에서는 이러한 문제점들을 해결하고 일반적인 상황에 적용시키기 위해 기존의 강화학습을 보다 일반화하여 초기에 환경 정보를 가지지 않은 상태로부터 불완전한 관측 정보와 보상을 바탕으로 상태와 정책을 동시에 학습시키는 모델을 제시한다.

II. 관련 연구

앞서 말한 바와 같이 기존 강화 학습은 상태의 수가 클 경우 문제가 발생하고, 환경이 완전히 관측 가능하지 않을 경우에는 그대로 적용할 수 없다. 이러한 문제를 해결하고 실제 문제에 적용하기 위해 여러 가지 방법들이 고안되어 왔다.

2.1 State Approximation

간단한 문제의 경우 상태들을 모두 테이블에 저장하고 각각의 상태에 대해 계산하는 것이 가능하다. 그러나 실세계문제의 경우 모든 상태를 테이블의 형태로 저장하는 것은 공간적으로 문제가 있고 실제 학습에 사용할 수 있는 데이터가 상태에 비해 현격히 부족하게 되어 (sparse data problem) 문제가 생기게 된다. 이러한 문제 해결을 위해 상태를 테이블에 그대로 저장하는 대신 함수 근사기(function

approximator)를 사용하는 방법이 많이 사용되어 왔다. 이는 신경망이나 결정 트리 등의 함수 근사기를 사용하여 상태 테이블을 근사하는 방법이다. 이러한 방법은 실제 TD-gammon 등에 사용되어 우수한 결과를 낳기도 했다. 하지만 함수 근사기를 사용하여 Q-Learning 등의 기존 강화 학습 알고리즘을 수행시에는 함수 근사기가 최종 학습될 목표를 근사할 능력이 있다고 해도 최적해로 수렴하는 것이 보장되지 않으며 최악의 경우 해에 수렴하지 않고 진동할 수도 있다[1].

2.2 State Division/Unification

실제 문제에선 상태들의 수가 매우 많이 존재하지만 그들 각각의 상태가 모두 별도의 저장 공간을 필요로 할 정도로 다른 경우는 드물다. 이러한 유사한 상태가 많기 때문에 유사한 상태를 묶음으로써 상태의 개수를 크게 줄이는 것이 가능하다.

이러한 상태 통합 방법은 두 가지 방향이 존재한다. 우선 초기에는 많은 상태로 시작해서 유사한 상태를 점점 통합하여 나가는 방법이 가능하고 반대로 초기의 하나의 상태에서 차이점이 발견되는 경우 상태를 나눠 나가는 방법이 가능하다. 전자의 경우 (Hierarchical RL) 환경 정보가 필요하고 초기에는 상태의 수가 많기 때문에 저장 공간의 감소에는 크게 도움이 되지 않으나 수렴속도측면에서의 향상이 있다. 후자의 경우에는 환경을 모를 경우에도 적용이 가능하다는 큰 장점이 있으며 저장 공간도 줄일 수 있는 장점을 가진다. Function approximator를 사용하는 경우는 approximator 내부에서 implicit하게 state의 조직화가 일어난다고 볼 수 있고, 위와 같은 경우는 explicit하게 조직화가 일어난다고 볼 수 있다. explicit하게 state를 유지하는 경우의 장점은 엄밀하게 상태를 통합할 경우 함수 근사기를 쓰는 것과 같이 최적해를 못 구하는 경우의 발생을 막을 수 있는 점에 있다.

2.3 Partially Observable Environment

기본적인 강화학습에서는 학습자가 Markov 성질을 가지는 환경에 대해 모두 관측할 수 있는 것을 전제로 한다. 많은 경우 이러한 가정은 성립하지 않는다. 환경 자체는 Markov 하지만 학습자에게 환경의 일부분만 알려지는 경우를 Partially Observablc Markov Decision Problem(POMDP)이라 한다. 이럴 경우 실제의 상태가 다른 경우에도 학습자에게는 같은 상태로 보이는 현상(perceptual aliasing)이 발생하여 이를 무시하고 바로 적용할 경우 결과가 불안정하다. POMDP의 경우 기존 강화학습의 deterministic한 행동이 확률적인 행동에 비해 임의로 나쁠 수 있다는 것이 알려져 있다[5].

이러한 POMDP에서의 강화학습의 적용시에는 기본적인 강화학습처럼 현재의 상태만을 가지고 행동 결정시 관측 못한 내부 상태를 무시하게 되어 성능이 떨어지게 된다. 이를 극복하기 위해 과거 취한 행동 등의 추가 정보를 사용하여 상태를 구분하게 된다. 모델이 주어졌을 경우에는 현재의 상태가 무엇인지를 추정하여 사용하는 방법(Belief state)으로 주어진 조건에서 최적의 해를 구할 수 있다. 환경이 주어지지 않았을 경우에는 환경을 학습하는 것이 필요해지는데 이는 앞서 다룬 방법으로 어느 정도 가능해진다.

III. 강화 학습 모델

앞서 말한바와 같이 기본적인 강화 학습 모델은 실제 문제에 적용하기에는 적합하지 못하다. 여기서는 기본적인 Q-Learning 모델을 소개하고 이를 단계적으로 확장하여 부분적으로 관측가능한 환경에서 상태와 정책을 학습하는 모델을 제시한다.

3.1 Q-Learning Model

Q-Learning 은 현존 강화 학습 중 대표적으로 쓰이는 방법으로써 시간 변화에 따른 적합도 차이를 학습에 이용하는 Temporal Difference 학습에 속한다. Q-Learning은 각 상태마다 행동의 적합성을 표시하는 Q값을 바로 학습하게 된다. 상태 s에서 행동 a를 택했을 경우 Q값은 다음과 같이 변화한다 여기서 α 값은 learning rate이고 γ값은 멀리 떨어진 state의 Q값이 영향을 미치는 discount factor 이다. s'는 a를 택한 결과 로 이동하는 다음 상태이고, a'는 다음 상태에서 가능한 행동이다.

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \quad (식1)$$

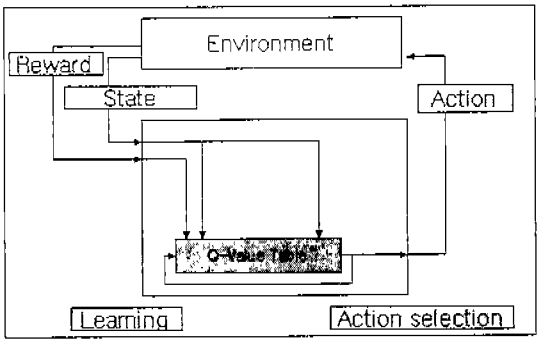

그림 1 . Q-Learning 모델

학습률 α를 잘 낮춰가면서 Q값에 기반한 확률적 정책이나 ε-greedy 정책을 사용할 경우 Q값은 수렴하고 수렴한 Q값에 greedy한 정책을 취하면 그 정책이 최적인 정책임이 알려져 있다.[3] 이러한 최적 수렴이 증명되어 있는 점과 사용하기 비교적 간단한 점이 Q-Learning의 큰 장점이지만 앞서 말한 바와 같이 실제 문제에 그대로 적용시에는 여러 가지 문제점이 있다.

3.2 Q-Learning Model with State Approximator

실제 문제에 Q-Learning을 적용하는데 있어서 가장 큰 문제는 상태의 수가 테이블 식으로 처리하기에는 너무 많은 것이다. 따라서 대부분의 경우 상태를 기본 Q-Learning처럼 table로 저장하지 않고 함수 근사기를 사용하여 저장하게 된다. 이 경우에는 올바른 결과를 얻지 못할 가능성이 있으나 저장 공간, 수렴 속도 등의 문제를 해결할 수 있으므로 많이 쓰인다. 이 경우에는 Q-Value를 저장하는 단계에서 Function Approximator에게 state-action-value를 input으로 주어 교사학습시키고, 읽는 단계에선 state-action을 입력으로 주고 출력으로 Q value를 얻는 방법으로 Q-Learning 구현이 가능하다. Function approximator는 supervised learning이 가능한 임의의 approximator가 사용될 수 있다. 상태를 explicit하게 통합시킬 경우도 function approximator를 사용하는 경우와 동일하게 볼 수 있다. 이 경우는 서로 다른 상태를 감지하고 나누거나 유사한 상태를 통합하는 과정이 필요하다. 두 경우 모두 앞서 설명한 Q-Learning 모델을 Table을 Approximator로만 대체하여 사용이 가능하다.

3.3 Q-Learning Model with State Estimation

불완전한 관측만 가능할 경우에는 앞서 말한 바와 같이 관측 가능한 결과를 바로 상태로 사용하게 되면 성능이 매우 떨어진다. 따라서 상태에 대한 정보를 더 얻기 위해서는 기존의 관측값등의 정보를 추가적으로 사용하여 현재의 상태를 인식할 필요가 있다. 추가적인 정보가 될 수 있는 것은 이제까지 학습자가 겪은 경험의 일부가 될 수 있다. 즉 관측 정보에 국소적 기억(Local memory)를 첨가하여 현재 상태의 불확실성을 해결하는데 도움을 줄 수 있다.

환경이 알려져 있는 경우에는 관측할 수 없는 환경의 상태에 대한 추정치를 계산하여 그것을 사용하여 Q-Learning을 수행할 수 있다. 환경이 알려져 있지 않은 경우 관측 결과로부터 내부 상태를 학습하면서 그 상태를 사용하여 학습과 정책 결정을 수행하게 할 수 있다. 이 경우는 앞서의 State approximation와 상태를 학습한다는 면에서 매우 유사하다.

3.4 Integration of State Approximation and State Estimation

앞서 살펴본 바와 같이 Approximator를 사용하여 상태를 저장하는 경우와 부분적으로 관측가능한 환경에서 상태를 학습하는 것은 큰 유사성을 가진다. 본 논문에서는 이 둘을 통합하여 부분적으로 관측가능한 환경에서 상태를 근사기를 사용하여 학습한다.

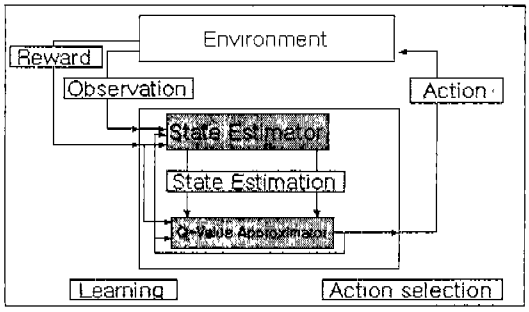

그림 2 .부분적으로 관측가능한 환경에서의 Q-Learning 모델

위의 두 모델을 그대로 통합할 경우의 모델은 그림 2와 같이 관측 결과로부터 상태를 추정한 후 추정된 상태(State estimation)을 실제 상태처럼 사용하여 Q-Value approximator를 학습시키는 것이 된다.

이 경우 State estimator에서는 explicit한 상태의 조직화가 일어나며 Approximator에서는 implicit한 상태의 조직화가 일어나게 된다. 만일 환경이 알려져 있다면 State estimator에서 실제 환경의 상태를 추정하는 것을 목표로 둘 수 있으나 환경이 알려져 있지 않은 경우에는 두 단계의 상태 조직화가 굳이 필요하지 않고 하나의 State Approximator만을 사용하여 상태의 추정과 조직화, 학습을 모두 처리할 수 있다. 이 경우의 모델은 다음과 같다.

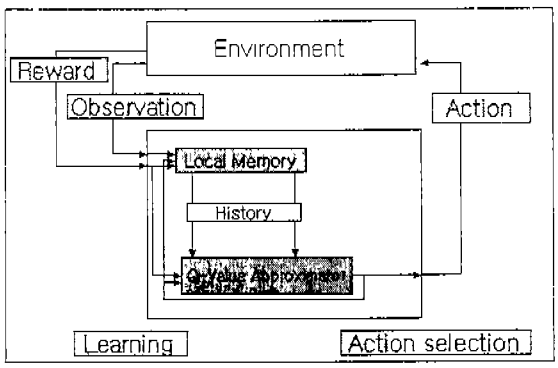

그림 3 . 통합된 Q-Learning 모델

3.5 Self-organizing state approximator

앞서 살펴본 일반화된 RL에서는 Q-value approximator에서 상태의 추정과 근사, 학습이 모두 행해진다. 이러한 Approximator에서 행해지는 일은 정책 결정과 정책/상태 학습의 두 가지이다. 정책 결정은 관측 결과들로부터 Q값을 출력하는 것이고 정책/상태 학습은 관측 결과, 행동, 보상으로부터 Q값을 수정해 나가는 것이다. 이러한 두 조건은 임의의 교사학습이 가능한 함수 근사기를 사용할 경우 만족이 가능하다. 즉 신경망, 결정트리 등이 사용될 수 있을 것이다.

본 논문에서는 근사기로 Evolutionary Algorithm 적인 관점에서 접근한 Self-organizing state approximator를 제시한다. 이 방법에서는 입력 공간에 Q-Value를 가지는 여러 개체들이 존재하게 된다. 환경 선택시에는 입력 정보에 근접해 있는 개체(혹은 개체들)의 정보를 사용하여 해당 입력의 Q-Value를 구하고, 반대로 Q-Value 학습 시에는 입력 정보에 근접해 있는 개체(혹은 개체들) 이 가지고 있는 Q-Value를 수정하여 학습을 수행하게 된다. 상태의 통합은 각 개체들간의 경쟁과 통합, 그리고 새로운 개체의 생성으로 이루어진다.

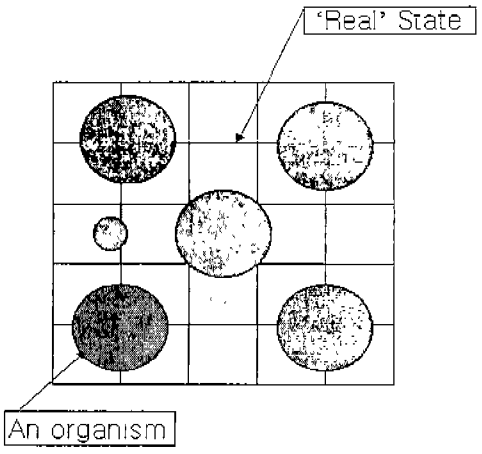

그림 4. 상태의 자기 조직화

완전히 관측 가능한 환경에서는 실제의 상태 공간에 개체들이 위치하게 할 수 있을 것이다. 이 경우 n차원의 상태 공간에서 특정 상태의 Q값을 얻기 위해서는 해당 상태에 해당하는 공간상의 점 위치에서 최단 거리에 있는 개체의 Q값을 사용하게 된다. 만일 환경이 부분적으로만 관측 가능할 경우 최근의 관측 값을 상태 공간으로 잡은 후 같은 방법을 적용할 수 있다. 이러한 개체들의 집합을 사용할 경우 극단적인 경우 개체와 상태를 1:1 대응시킬 경우 임의의 상태 공간의 Q-value를 근사하는 것이 가능하다. 반면 이러한 경우 이러한 개체를 쓰는 이점이 없기 때문에 개체의 수를 줄이면서 목표하는 상태에 적응시키는 것이 필요하다. 이를 위해서는 각 개체를 단순한 Node가 아닌 개체로 보고 각 개체가 자신의 생존을 위해기 적응해 나가는 개념을 도입하였다. Genetic Algorithm에서와 같이 환경과의 적합도에 따라 각 개체는 도태되거나 증식하게 된다. 이 방법은 환경이 시간에 따라서 변할 때에 환경에 적응해 나갈 수 있는 잠재력을 가진다.

IV. 실험 및 결과

실험은 부분적으로 관측 가능한 미로 탐색 문제로 설정하였다. 목표는 아래와 같은 환경에서 최단 거리의 경로를 찾아내는 것이다. 최단 경로를 찾기 위해서 보상은 움직일 때마다 -1, 목적지에 도달할 경우 100을 주게 하였다. 미로는 기존의 GridWorld실험처럼 빈 공간과 찬 공간으로 이루어져 있지 않고 Micro Mouse에서 사용하는 미로와 같이 셀 사이에 벽이 놓여 있는 형식으로 설정하였다. 이를 통해 격자 개수에 비해 보다 복잡한 미로의 설정이 가능하고 차후에 실제 Micro mouse 용 미로의 적용도 가능하게 하였다.

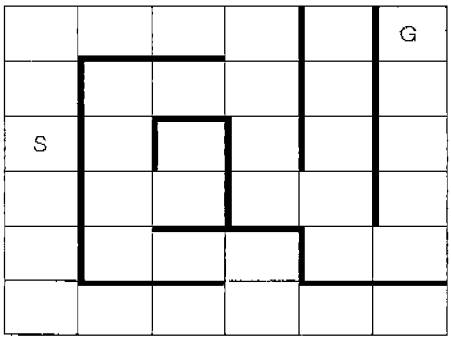

그림 5 실험에 사용된 미로

실험은 위의 환경에서 세 단계로 나누어 진행하였다. 우선 처음에는 상태가 완전히 관측 가능한 환경에서 테이블을 사용한 Q-Learning을 사용하여 학습시켰다. 두 번째로는 테이블을 대신하여 자기조직화 상태 근사기를 사용하여 상태와 정책을 학습시켰다. 마지막으로는 부분적인 상태 정보만 주어진 상태에서 상태와 정책을 학습시켰다.

4.1 Q-Learning

Q-Learning으로 학습시킨 결과 아래 그림과 같이 최적의 해를 무리 없이 찾았다. 상태는 모두 관측 가능하고 6*6 격자의 각 위치를 상태로 잡았다. ε-greedy 정책을 사용하였고 ε 초기 값은 0.3 , α값은 0.5, γ값은 0.7로 잡은 뒤 시간에 따라 linear하게 낮춰주었다.

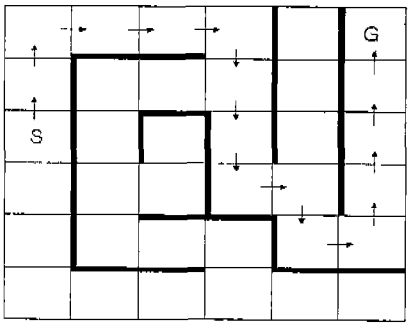

그림 6 Q-Learning으로 구한 최적해

4.2 Q-Learning with Self-organizing states

A-Life based인 State approximator 와 Q-Learning을 사용하여 같은 실험을 반복하였다. 상태 정의는 앞서와 동일하게 하였다. 특정 최단 거리에 있는 개체가 가진 정보를 그 상태의 정보로 사용한다. 같은 개체가 커버하는 범위에서 다른 관측값이 관측되었을 경우 새로운 개체가 생성된다. 미로 문제의 특성상 환경이 변하지 않고 상태의 풍경이 복잡하기 때문에 개체 적응은 사용하지 않았다. 실험한 결과 앞서 구한 최적해를 구할 수 있었으며 상태의 수는 원래의 25% 만큼 감소하였다.

4.3 Q-Learning with Self-organizing states in Partially Observable Environment

실험의 마지막 단계로써 두 번째의 실험 설정에 추가하여 환경을 부분적으로만 관측 가능하게 하였다. 즉 현재 위치(상태)는 학습자에게 알려지지 않으며 단지 현재 위치에 인접해 있는 벽의 상태만 알려진다.

이러한 경우의 문제를 해결하기 위해 최근 N 스텝 동안의 움직임을 기억하는 Local memory를 사용하였다. 즉 Local memory와 현재 인접한 상태를 상태로 사용하여 두 번째 실험에서 사용한 방법으로 학습을 시켰다. N스텝 동안의 모든 움직임의 조합을 상태로 줄 경우 상태 공간이 매우 복잡하고 커지기 때문에 N스텝 동안의 상대적인 움직임의 합을 상태로 사용하였다. 실험 결과 N이 아주 작을 경우에는 제대로 된 결과가 나오지 않았으나 N이 20 정도로 놓았을 경우 최적해를 구할 수 있었다. 초기의 탐색 단계에서는 목표 지점에 도달하는데 걸리는 기리는 그림 7과 같이 최적해의 경우에 비해 상당히 클 수 있다. 따라서 20 정도의 N의 사용으로도 Local memory를 사용함으로써 이익을 얻었다고 할 수 있다.

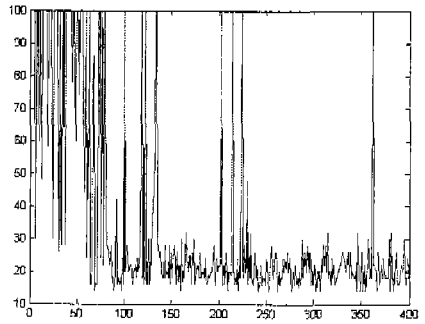

그림 7. 초기 n에 대해서 cost의 수렴

V. 결론 및 연구 방향

본 논문에서는 기존 강화 학습을 실제 문제에 적용하는 데 있어서 주된 어려움이었던 상태 수의 폭증, 알려져 있지 않은 모델, 부분적으로만 관측 가능한 환경 문제를 모두 해결 가능한 일반적인 강화 학습 모델을 제시하였고 간단한 실험을 통해 확인하였다. 차후에는 본격적인 개체 레벨의 적응을 도입하여 GA 및 A-Life 분야에서의 연구 결과를 강화 학습에 적극적으로 도입하는 것과 보다 일반적이고 큰 문제에의 적용을 목표로 하고 있다.

감사의 글 : 본 연구는 BK21 프로젝트에 의해 일부 지원 받았습니다.

VI. 참고문헌

[1] Boyan,J.A., and Moore,A.W. "*Generalization in reinforcement learning:Safely approximating the value function.*" In Advances in neural Information Processing Systems, volume 7. 1995.

[2] Moore,A.W. "*The parti-game algorithm for variable resolution reinforcement learning in multi-dimensional state-spaces.*" In Advances in Neural Information Processing Systems 6, 1994.

[3] Watkins, C.K.C.H., and Dayan,P. "*Q-learning*".Machine Learning 8, 1992.

[4] A.McCallum. "*Reinforcement Learning with Selective Perception and Hidden state.*" PhD thesis, Univ. Rochester, 1995.

[5] L.P.Kaelbling, M.Littman, and A.Cassandra. "*Planning and acting in partially observable stochastic domains.*" Artificial Interlligence, 101, 1998.