# Planejamento Probabilístico

MAC5701 – Tópicos em Ciência da Computação

Felipe Werndl Trevizan
Orientadora: Leliane Nunes de Barros

**Resumo**

ABSTRACT

## 1 Introdução

Em Inteligência Artificial, o problema de selecionar ações em ambientes dinâmicos que não sejam completamente previsíveis ou observáveis pode ser visto como o desenvolvimento de controladores (Figura 1) que mapeiam seqüências de observações em ações para atingir uma determinada meta. A construção desses controladores pode ser feita através de três abordagens:

- **programação**, onde o controlador é programado manualmente através de uma linguagem procedimental de alto-nível,
- **planejamento**, no qual o controlador é derivado automaticamente através de uma descrição correta das ações, dos sensores e das metas desejadas,
- **aprendizado de máquina**, onde o controlador é derivado através de experiências supervisionadas sobre um conjunto de problemas de teste.

Todas essas abordagens citas apresentam tanto vantagens como limitações, e exemplos de suas utilizações, respectivamente, são: programação de sistemas especialistas em robôs; teoria de controle e planejamento em Inteligência Artificial (IA); e por último generalização de modelos e aprendizagem por tentativa e erro.

Em planejamento em IA, técnica usada nesse texto, o controlador derivado recebe o nome de *plano* solução. Um plano pode ser simplesmente uma seqüência

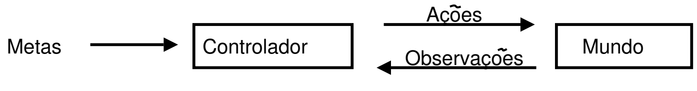


Figura 1: Diagrama de funcionamento de um controlador. Um controlador deve mapear uma seqüência de observações em ações para atingir uma determinada meta.

de ações que devem ser executadas, recebendo o nome de plano de malha aberta (*open-loop plan*), ou uma função cuja entrada são as percepções do ambiente e a saída é a ação que deve ser executada naquele instante. Esse último tipo de plano, com retroalimentação, recebe o nome de plano de malha fechada (*closed-loop plan*) e tem como principal exemplo os planos de contingência, que são planos com operadores: `if`, `while` e `until`.

Uma das modelagens mais simples para problemas de planejamento é o modelo determinístico de estado, capaz de modelar problemas cujas ações são determinísticas e o ambiente é completamente observável. Esse modelo, detalhado na seção 1.1, representa os problemas da área de planejamento determinístico que possuem como solução planos de malha aberta.

Nesse texto será apresentado um arcabouço (*framework*) para modelar e resolver problemas de planejamento probabilítico, uma extensão do planejamento determinístico para ambientes estocásticos. Esse arcabouço será composto por modelos matemáticos para tomada de decisões capazes manipular ações determinísticas e probabilíticas em ambientes completamente, parcialmente e não-observáveis. Além disso, também serão definidos algoritmos para resolver tais estruturas obtendo planos de malha fechada.

## 1.1 Planejamento determinístico

Um problema de planejamento determinístico pode ser descrito através de um modelo de estado, composto por um conjunto finito de estados $\mathcal{S}$, um conjunto de ações $\mathcal{A}$ e uma função de transição $\mathcal{T}$ que descreve como ações mapeiam um estado em outro. Formalmente um problema de planejamento determinístico é definido pela tupla $\langle \mathcal{S}, s_0, \mathcal{A}, \mathcal{T}, \mathcal{C}, \mathcal{G} \rangle$, no qual:

- $\mathcal{S}$ é um conjunto finito de estados possíveis para o sistema,
- $s_0 \in \mathcal{S}$ é o estado inicial do sistema,
- $\mathcal{A}$ é o conjunto de ações do sistema e $\mathcal{A}(s)$ representa as ações aplicáveis no estado $s$,
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ é a função de transição representando o estado do sistema após executar a ação $a \in \mathcal{A}(s)$ no estado $s \in \mathcal{S}$,
- $\mathcal{C} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ é uma função representando o custo de aplicar a ação $a$ no estado $s$,
- $\mathcal{S}_\mathcal{G} \subseteq \mathcal{S}$ é o conjunto de estados meta do sistema.

A solução de um problema de planejamento determinístico é uma seqüência de ações $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ chamada de *plano solução*, que nesse caso é um tipo de plano malha aberta. Esse plano solução gera a seguinte seqüência de estados (trajetória), $\langle s_0, s_1 = \mathcal{T}(s_0, a_0), \ldots, s_n = \mathcal{T}(s_{n-1}, a_{n-1}) \rangle$, onde $\forall i, 0 \leq i < n, a_i \in \mathcal{A}(s_i)$ e $s_n$ é um estado meta, ou seja, $s_n \in \mathcal{S}_\mathcal{G}$. Esse plano é *ótimo* se o custo total da solução, $\sum_{i=0}^{n-1} \mathcal{C}(s_i, a_i)$, é mínimo.

Um caso particular de planejamento determinístico é o planejamento clássico, que além de possuir ações determinística, espaço de estados ($\mathcal{S}$) finito, ainda possui a seguintes restrições:

- o ambiente é totalmente observável (onisciência), ou seja, o agente sempre tem acesso a todas as propriedades do ambiente, independente do estado ou de probabilidades,
- o ambiente é modificado apenas pelas ações do agente, ou seja, não existem outros agentes no ambiente nem a ocorrência de ações exôgenas,
- o planejador trata apenas com metas restritas, especificadas por $S_G$, ou seja, não é possível incluir problemas de otimização na meta, nem declarar uma seqüência de estados que devem ser visitados ou evitados,
- um plano solução é uma seqüência finita de ações totalmente ordenadas,
- ações não possuem duração, ou seja, elas são instantâneas e
- o custo de todas as ações é 1, ou seja, $\mathcal{C}(s, a) = 1, \forall s \in \mathcal{S}$ e $a \in \mathcal{A}(s)$, assim um plano solução ótimo em planejamento clássico é o plano solução com o menor número de ações ($\sum_{i=0}^{n-1} \mathcal{C}(s_i, a_i) = \sum_{i=0}^{n-1} 1 = n$)

O estado da arte para a solução de problemas de planejamento clássico envolvem tanto planejadores baseados em busca no espaço de estados, através do uso de estruturas de dados compactas e heurísticas independentes de domínio [Bonet and Geffner, 2001a], quanto planejadores baseados em busca no espaço de planos. Essa última forma de busca ser decomposta em busca por rede de tarefas [Nau et al., 2003] e busca por safistação de restrições [Kautz and Selman, 1999].

# 2 Modelagem de problemas de planejamento probabilítico

Planejamento probabilístico difere do planejamento determinístico em dois pontos principais: primeiro, o efeito das ações são probabilísticos, ou seja, ao executar uma ação, o agente não sabe *a priori* o estado resultante do sistema, e segundo, o ambiente nem sempre será completemente observável, ou seja, o agente pode não saber exatamente qual o estado atual do sistema, mas sim qual o conjunto de estados possíveis para o sistema naquele instante.

Nessa seção serão apresentados os modelos matemáticos do arcabouço para solucionar problemas de planejamento probabilísticos. Todos os modelos que serão apresentados são *markovianos*, ou seja, respeitam a propriedade de Markov, que nesse caso pode ser enunciada da seguinte forma: o estado do sistema no tempo $t+1$ e a recompensa esperada no tempo $t+1$ dependem apenas do estado do sistema no tempo $t$ e da ação executada no tempo $t$.[1] Dessa forma,

[1] Esses modelos podem ser adaptados para cadeias de Markov de grau $k$, para qualquer $k$ finito. Nesse caso a propriedade de deduzir propriedades no tempo $t+1$ dependerá da seqüência de propriedades no tempo $\langle t-k+1, t-k+2, \ldots, t \rangle$

qualquer informação sobre os estados $s_0, s_1, \ldots, s_{t-1}$ podem ser ignoradas para tomar decisões sobre o estado $t+1$.

## 2.1 Completamente observável

Entre as hipóteses de conhecimento do agente sobre os ambientes (completamente, parcialmente e não observável), a mais simples de ser modelada é de observação completa (onisciência), por isso o modelo apresentado será usado como base para os outros casos.

Um processo markoviano de decisão (*Markov Decision Process*), que será chamado de MDP, é usado para modelar problemas de planejamento probabilístico em ambientes completamente observáveis. No arcabouço dos MDPs é suposto que, apesar da incerteza dos efeitos das ações executadas, nunca há incerteza sobre o estado atual do sistema, pois a percepção do ambiente é completa e correta. Assim, o agente só saberá qual é o estado resultante da execução de uma ação probabilística depois de receber as percepções do ambiente.

Um MDP pode ser descrito pela tupla $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{C}, \mathcal{R}, \mathcal{G} \rangle$:[2]

- $\mathcal{S}$ é o espaço de estados finito[3], onde será usada a hipótese de Markov, ou seja, cada estado possui todas as informações possíveis para determinar o efeito de qualquer ação,
- $\mathcal{A}$ é o conjunto de ações e $\mathcal{A}(s)$ representa as ações aplicáveis no estado $s$,
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ é a função de transição de estado, que para cada estado $s \in \mathcal{S}$ e ação $a \in \mathcal{A}(s)$ gera um distribuição de probabilidade sobre o espaço de estados $\mathcal{S}$. Será usada a notação $\mathcal{T}(s, a, s') = P(s'|s, a)$ para representar a probabilidade do sistema estar no estado $s'$ após ser executada a ação $a \in \mathcal{A}(s)$ no estado $s \in \mathcal{S}$,
- $\mathcal{C} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ é uma função representando o custo de aplicar a ação $a$ no estado $s$,
- $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ é uma função representando a recompensa recebida pelo agente por estar no estado $s$,
- $\mathcal{G} \subseteq \mathcal{S}$ é o conjunto de estados meta.

Para calcular a recompensa esperada durante o tempo de funcionamento do agente será usada uma função de valoração $\mathcal{V} : \mathcal{H}_{\mathcal{S},\mathcal{A}} \to \mathbb{R}$. Essa função mapeia cada elemento do conjunto de históricos de execuções válidas ($\mathcal{H}_{\mathcal{S},\mathcal{A}}$), isso é uma seqüência da forma $\langle (s_0, a_0), (s_1, a_1), \ldots, (s_{n-1}, a_{n-1}), s_n \rangle, s_i \in \mathcal{S}, 0 \leq i \leq n$ e $a_j \in \mathcal{A}, 0 \leq i < n$, à pontuação recebida pelo agente ao percorrer esse histórico.

A função de valoração está intrinsecamente ligada na quantidade de passos máximos, $n$, que o agente pode ter em seu histórico ($\|h\| \leq n, h \in \mathcal{H}_{\mathcal{S},\mathcal{A}}$). Caso

[2] A definição usada nesse texto é bemgenérica, englobando as definições de [Bonet and Geffner, 2001b, Kaelbling et al., 1996] e muito similar a [Boutilier et al., 1999].

[3] Apesar do espaço de estados $\mathcal{S}$ ser finito, é possível adapatar a definição de MPD, bem como seus métodos de resolução, para trabalhar com um espaço de estado infinito contável.

$n$ seja finito, chamado de MDP de horizonte finito, $\mathcal{V}$ pode ser calculado da seguinte maneira:

$$\mathcal{V}(h) = \sum_{t=0}^{n-1}(\mathcal{R}(s_t) - \mathcal{C}(s_t, a_t)) + \mathcal{R}(s_n).$$

Um MDP de horizonte finito por ser visto como um problema onde o agente possui algum recurso finito, como por exemplo energia, bateria ou combustível, que limita a quantidade de ações que ele poderá executar no ambiente. Como raramente é conhecido a quantidade passos $n$ que o agente pode executar, é necessário definir $\mathcal{V}$ quando o $n$ é infinito, ou seja, para um MDP de horizonte infinito.

Como a soma da pontuação de cada passo de um histórico infinito $h$ pode divergir, é necessário introduzir um *fator de desconto* no cálculo da pontuação total do agente, gerando um limitante superior para a recompensa. Dessa maneira, $\mathcal{V}$ para um MDP de horizonte infinito é definido por

$$\mathcal{V}(h) = \sum_{t=0}^{\infty}\gamma^t(\mathcal{R}(s_t) - \mathcal{C}(s_t, a_t)), \text{ onde } 0 < \gamma < 1,$$

tendo como limitante superior $\frac{\mathcal{R}_{max} - C_{min}}{1-\gamma}$, onde $\mathcal{R}_{max}$ é a maior recompensa que o agente pode receber e $C_{min}$ é o menor custo de uma ação. Nesse modelo, o fator de desconto $\gamma$ reflete a preferência do agente em executar ações mais lucrativas no início ($\gamma$ próximo de 0) ou no decorrer ($\gamma$ próximo de 1) do seu histórico.

Uma vez definido o tipo de horizonte do MDP, é possível procurar uma solução que seja ótima, ou seja, que maximize a pontuação do agente. Como o efeito das ações são probabilísticos, a solução de um MDP deve atribuir a cada trajetória, uma seqüência de estados válida, uma probabilidade. Isso porque depois de executar uma ação, o agente não sabe *a priori* qual será o estado resultante. Assim, uma solução $\pi$ de um MDP é avaliada pela sua recompensa esperada (esperança ?), que é:

$$\mathcal{V}_\pi = \sum_{h \in \mathcal{H}_{\mathcal{S},\mathcal{A}}} \mathcal{V}(h)P(h|\pi).$$

A maneira mais usada para representar soluções para MDPs, que será adotada nesse texto, são as *políticas*. Uma política $\pi$ é um plano de malha fechada representado por uma função que mapeia para cada estado do sistema $s \in \mathcal{S}$ uma ação $a \in \mathcal{A}$ que deve ser executada em $s$. Uma política define a probabilidade de cada trajetória possível do sistema:

$$P(\langle s_0, s_1, \ldots, s_{n-1}, s_n\rangle|\pi) = \sum_{i=1}^{n} P(s_i|s_{i-1}, \pi(s_{i-1})) = \sum_{i=1}^{n} T(s_{i-1}, \pi(s_{i-1}), s_i).$$

Uma política pode ser *estacionária*, quando ela não se altera com o tempo, ou seja, a melhor ação a ser executada no estado $s$ é sempre a mesma. Caso contrário, quando a melhor ação pode se alterar com o tempo, a política é chamada de *não-estacionária* e é representada pela função $\pi(s,t), s \in \mathcal{S}$ e $t \in \mathbb{R}$, onde $s$ é o estado atual e $t$ é a quantidade de ações que o agente ainda pode executar.

A solução de MDPs de horizonte finito são, em geral, políticas não-estacioná rias, pois o agente tende a agir de forma diferente nos primeiros passos, tentando maximizar a recompensa global, do que no seu último passo, onde maximizar a recompensa global é equivalente a maximizar a recompensa adiquirida executando apenas uma ação. Assim, uma política não-estacionária para um MDP de horizonte $T$, pode ser vista como uma coleção de $T$ políticas estacionárias, uma para cada ponto de decisão do agente.

### 2.1.1 Planejamento determinístico como um MDP

Um problema de planejamento determinístico, definido na seção 1.1, $\langle \mathcal{S}, s_0, \mathcal{A}, \mathcal{T}, \mathcal{C}, \mathcal{G} \rangle$ pode ser re-escrito na forma de um MDP de horizonte finito $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}', \mathcal{C}, \mathcal{R}, \mathcal{G} \rangle$, onde:

- $\mathcal{T}'(s,a,s') = \begin{cases} 1 & \text{se } \mathcal{T}'(s,a) = s' \\ 0 & \text{caso contrário} \end{cases}$, logo esse MDP é determinístico, pois ao executar uma ação, apenas um estado possuirá probabilidade maior que 0, e essa probabilidade será 1, assim o agente sabe exatamente quais são os efeitos de suas ações.

- $\mathcal{R} = 0$, pois em um problema de planejamento determinístico não pode haver preferência entre estados.

A solução do problema de planejamento determinístico $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ é obtida através da política $\pi$ e segue a forma $\langle \pi(s_0, T), \pi(s_1, T-1), \ldots, \pi(s_{T-1}, 1) \rangle$. Note que não é necessário realizar sensoriamento do ambiente para encontrar $s_i, 0 < i < T$, pois a execução de uma ação é determinísca devido a definição de $\mathcal{T}$.

## 2.2 Parcialmente observável

Para modelar um ambiente parcialmente observável, o arcabouço dos MDPs será extendido para permitir que o sensoriamento do mundo seja probabilístico. Assim, como na função de transição, serão definidas probabilidades para o agente receber uma determinada percepção após executar uma ação $a$ gerando um estado $s$.

Esse novo modelo matemático recebe o nome de processos markovianos de decisão parcialmente observáveis (*Partial Observable Markov Decision Process*) e serão chamados de POMDPs. Formalmente um POMDP é definido pela tupla $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{C}, \mathcal{R}, \mathcal{G}, \Omega, \mathcal{O} \rangle$:

- $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{C}, \mathcal{R}, \mathcal{G} \rangle$ descreve um MDP,

- $\Omega$ é um conjunto finito de observações possíveis que o agente pode receber do ambiente, e
- $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \Pi(\Omega)$ é a função de observação, que para cada ação $a \in \mathcal{A}$ e cada estado resultante $s \in \mathcal{S}$, devolve um distribuição de probabilidade sobre o espaço de observações possíveis ($\Omega$). Será usada a notação $\mathcal{O}(s, a, o) = P(o|s, a)$ para a probabilidade de observar $o \in \Omega$ ao aplicar a ação $a \in \mathcal{A}$ resultando no estado $s \in \mathcal{S}$.

Como a percepção do ambiente não é mais determinística, dificilmente o estado do sistema poderá ser determinado. Por isso será usado do conceito de *estado de crença*, que representa os estados que o agente considera possíveis para o sistema naquele instante. Um estado de crença $b$ é uma distribuição de probabilidade sobre o espaço de estados $\mathcal{S}$ e $b(s) = P(s|b)$ representa a probabilidade do sistema realmente estar no estado $s$ quando o estado de crença é $b$.

Como nos MDPs, a execução de uma ação muda a representação interna do agente sobre o ambiente, que nos POMDPs é o estado de crença. Assim, ao executar uma ação $a$ e receber a percepção $o$, o novo estado de crença $b'$ é a expansão de $b$, onde deve ser levado em conta todos os possíveis efeitos de $a$ sobre todos os estados $s \in \mathcal{S}$ e as probabilidades de observar $o$ em $s$. A probabilide do sistema estar realmente no estado $s'$ após executar a ação $a$ e observar $o$ no estado de crença $b$ é:

$$
\begin{aligned}
b'(s') &= P(s'|b, a, o) && (1)\\
&= \frac{P(o|b, a, s')P(s'|b, a)}{P(o|b, a)} && (2)\\
&= \frac{P(o|a, s')P(s'|b, a)}{\sum_{s \in \mathcal{S}} P(o|a, s)P(s|a, b)} && (3)\\
&= \frac{\mathcal{O}(s', a, o)P(s'|b, a)}{\sum_{s \in \mathcal{S}} \mathcal{O}(s, a, o)P(s|b, a)}. && (4)
\end{aligned}
$$

No conjunto de equações acima $b'(s')$ representa a probabilidade do sistema estar no estado $s'$ dado que o estado de crença do agente era $b$ e foi executada a ação $a$ e observado $o$ (1). A equação 2 usa a regra ??? da teoria básica de probabilidade, onde $P(s'|b, a)$ representa a probabilidade do sistema estar no estado $s'$ após executar a ação $a$ no estado de crença $b$ e $P(o|b, a)$ a probabilidade de observar $o$ após executar $a$ em $b$. A equação 3 usa o fato da probabilidade de observar uma propriedade do ambiente ser independente do estado de crença do agente e $P(o|b, a)$ ser a probabilidade conjunta de observar $o$ em $s$ após executar $a$ e de atingir o estado $s$ após executar $a$ no estado de crença $b$.

Na equação final (4) a probabilidade de atingir um estado $s$ após executar a ação $a$ no estado de crença $b$ não foi expandida. Essa probabilidade, $P(s|a, b)$, tem como identidade $\sum_{s' \in \mathcal{S}} P(s|b, a, s')P(s'|b, a) = \sum_{s' \in \mathcal{S}} T(s', a, s)b(s')$, ou seja, a probabilidade de estar em um estado $s'$ no estado de crença $b$ e atingir o

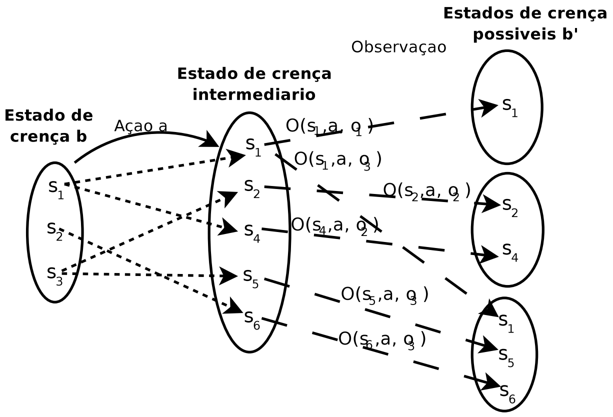


Figura 2: Processo de atualização do estado de crença do agente. A primeira oval é o estado de crença do agente, onde será aplicado a ação $a$. A segunda oval é um estado de crença intermediário que contém todos os estados atingíveis ao aplicar $a$ em $b$. As setas pontilhadas indicam quais estados em $b$ foram responsáveis pela criação do estado de crença intermediário. A última coluna de ovais representam os estados de crença possíveis ao aplicar $a$ em $b$. As setas tracejadas representam as observações responsáveis pela permanência dos estados em $b'$.

estado $s$ executado a ação $a$. Essa substituição é válida tanto no denominador quanto no numerador de 4.

A Figura 2 ilustra um processo de atualização do estado de crença do agente em duas etapas. O primeiro passo, representado pelas duas primeiras colunas de ovais, recebe um estado de crença $b$ e uma ação $a$ que será executada e gera um estado de crença intermediário, que contém todos os estado atingíveis a partir de $b$. Nesse estado de crença intermediário, a probabilidade de um estado $s$ é $P(s|a,b) = \sum_{s' \in \mathcal{S}} T(s', a, s)b(s')$, ou seja, é um pré-processamento para armazenar alguns valores que serão usados na equação 4.

O segundo passo, representado pela segunda e terceira coluna de ovais, usa a observação $o$ recebida para discriminar quais os estados possíveis para o novo estado de crença $b'$. Assim como em um MDP, onde é possível prever o efeito de uma ação através de uma distribuição de probabilidade sobre o espaço de estados $\mathcal{S}$, em um POMDP é possível prever o efeito de uma ação através de uma distribuição de probabilidade sobre os subconjuntos dos estados contidos no estado de crença intermediário que foi gerado. A probabilidade de um estado $s$ em qualquer estado de crença possível $b'$ é definido pela equação 4, onde o denominador deve ser calculado para cada $b'$ e $P(s|a,b)$ já é conhecido.

Esse método apresentado tem complexidade $O(\|\mathcal{S}\|^2)$, pois a primeira fase do algoritmo calcula uma somatória sobre o espaço de estados $\mathcal{S}$ para cada estado $s$, enquanto a segunda fase é linear dado a observação $o$.

### 2.2.1 Não observável

- Explicar como o não determinísmo fica determinístico 'as probabilidades de transição foram incorporadas ao modelo de estados (ED)'
- NOMDP -> Planejamento Determinístico

# 3 Algoritmos para solução de MDPs

- Iteração de Valor
- Iteração de política
- Programação dinâmica
- RTDP

## 3.1 Algoritmos para solução de POMDPs

- RTDP para POMDPs

# Referências


[Bonet and Geffner, 2001a] Bonet, B. and Geffner, H. (2001a). Heuristic search planner 2.0. *AI Magazine*, 22(3):77–80.

[Bonet and Geffner, 2001b] Bonet, B. and Geffner, H. (2001b). Planning and control in artificial intelligence: A unifying perspective. *Applied Intelligence*, 14(3):237–252.

[Boutilier et al., 1999] Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94.

[Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1996). Planning and acting in partially observable stochastic domains. Technical Report CS-96-08.

[Kautz and Selman, 1999] Kautz, H. and Selman, B. (1999). Unifying SAT-based and graph-based planning. In Minker, J., editor, *Workshop on Logic-Based Artificial Intelligence, Washington, DC, June 14–16, 1999*, College Park, Maryland. Computer Science Department, University of Maryland.

[Nau et al., 2003] Nau, D. S., Au, T. C., Ilghami, O., Kuter, U., Murdock, W., Wu, D., and Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, 20:379–404.