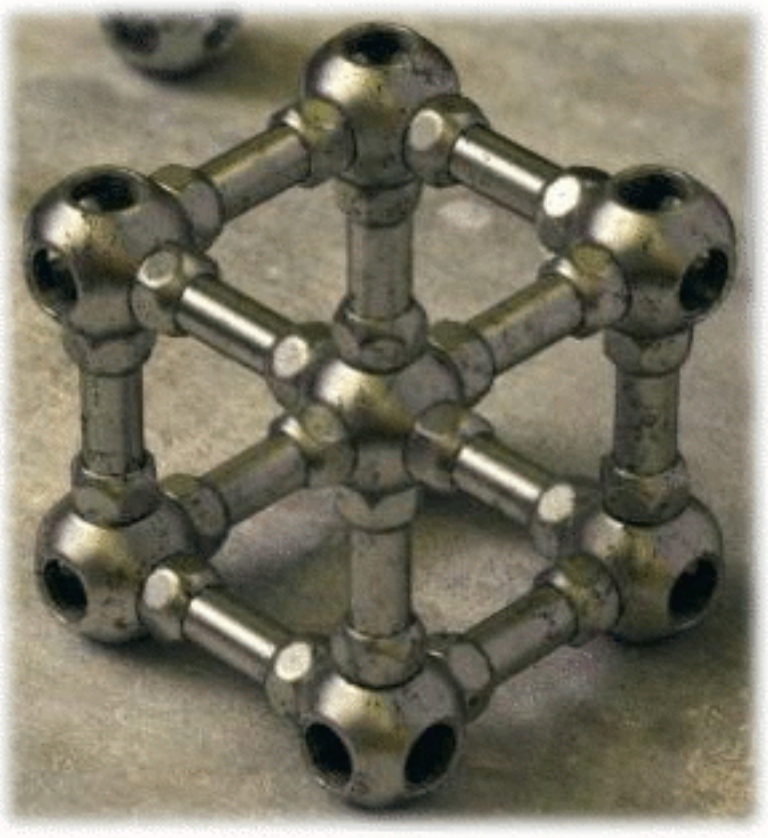

Gerando Políticas Não Dominadas em Processos Markovianos de Decisão [1]

Author(s):
Valdinei Freire da Silva
Anna Helena Reali Costa

[1]This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

Gerando Políticas Não Dominadas em Processos Markovianos de Decisão

Valdinei Freire da Silva
Laboratório de Técnicas Inteligentes
Escola Politécnica - USP
São Paulo, SP, Brasil
valdinei.freire@gmail.com

Anna Helena Reali Costa
Laboratório de Técnicas Inteligentes
Escola Politécnica - USP
São Paulo, SP, Brasil
anna.reali@poli.usp.br

***Resumo*—Processos Markovianos de Decisão é um formalismo utilizado para modelar problemas de decisões seqüenciais em ambientes não deterministas. Quando se possui uma descrição completa do processo, existem algoritmos tradicionais na literatura que podem encontrar políticas ótimas de ação. As preferências de um usuário são programadas por meio de uma função recompensa que modelam como as políticas devem ser avaliadas. Por outro lado, quando o usuário é leigo, algum sistema deve auxiliá-lo na definição de tal função representando as preferências do usuário na função recompensa. Esses sistemas consideram as opções de decisões disponíveis, nesse caso políticas, para guiar o processo de mapeamento das preferências do usuário nas funções recompensas. Neste trabalho apresenta-se um algoritmo para definir o conjunto de políticas não dominadas que é o conjunto mínimo de políticas que necessitam ser contempladas na tomada de decisão. Experimentos em um ambiente simulado de robótica móvel mostra a eficácia de tal algoritmo para reduzir a quantidade de políticas a serem consideradas.**

***Palavras-Chave* - Processo Markoviano de Decisão; Extração de Preferências; Política Não Dominada**

I. INTRODUÇÃO

Uma questão importante ao delegar a tomada de decisão a um sistema inteligente é como fazer com que tal agente satisfaça as decisões de seu projetista, ou, em muitos casos, de seu usuário. A área de Extração de Preferências (EP) preocupa-se em configurar sistemas inteligentes construídos de forma genérica para que satisfaçam as preferências de usuários específicos [1], [2].

Os métodos de EP baseiam-se em um processo de perguntas e respostas, sob a suposição de que as respostas às perguntas refletem características das preferências do usuário. Não é incomum que tais métodos levem em conta as opções disponíveis para tomada de decisão, limitando o processo de perguntas e respostas ao mínimo necessário para tomar uma decisão ótima, ou quase-ótima [3], [4], [5].

O espaço de opções pode ser considerado em sua totalidade, ou apenas alguma amostra de tal espaço. Enquanto a segunda opção diminui o tamanho do problema, ela exige que exista uma medida de distância entre as opções, e tal distância nem sempre é fácil de ser determinada.

Um arcabouço comum para tomadas de decisão é o Processo Markoviano de Decisão (MDP - *Markovian Decision Process* [6]), nesse é considerada uma seqüência de situações que são direcionadas por ações escolhidas pelo sistema e um componente estocástico. Dessa forma, o valor de uma ação depende das ações tomadas no futuro, implicando que decisões devem ser consideradas em conjunto. Em um MDP tais decisões são modeladas por políticas que definem uma ação a ser tomada para cada situação possível. A qualidade de uma política baseia-se em recompensas, que são avaliações parciais, locais e escalares que devem ser almejadas se positivas e evitadas quando negativas.

Embora no formalismo aplicado a um MDP uma recompensa seja modelada por uma função que relaciona um escalar a cada par estado-ação, é comum considerar uma função sobre atributos, que são conceitos mais abstratos: colisões em paredes, alcance de um determinado alvo, coletas de algum material, passagem do tempo, energia gasta, etc. Esse tipo de abstração diminui a quantidade de parâmetros a serem definidos para especificar uma função recompensa. Ainda, esta abstração permite especificar em poucos valores o resultado obtido ao executar uma determinada política, ou o resultado esperado de uma determinada política.

A quantidade de políticas possíveis cresce exponencialmente com a quantidade de estados possíveis, impossibilitando a enumeração de todas políticas. Por outro lado, embora não exista uma noção de vizinhança entre políticas quando comparando ações, pode-se estabelecer tal noção de vizinhança nos resultados obtidos, estabelecendo uma noção de distância no espaço de atributos.

Neste artigo apresenta-se um algoritmo para enumerar políticas não dominadas segundo um erro pré-definido. Esse algoritmo permite enumerar um conjunto de políticas de tal modo que, independentemente da função recompensa escolhida, a política ótima não pode ser melhor que um determinado erro.

Na seção II é apresentado o formalismo de MDPs, enquanto na seção III são apresentados alguns conceitos de EP. Na seção **IV** é apresentado um algoritmo para solucionar MDPs com horizonte finito. Nas seções **V** e **VI** são apresentados os resultados teóricos deste trabalho, enquanto na seção **VII**, os resultados práticas. Por último, na seção **VIII** apresenta-se uma breve conclusão.

II. Processos Markoviano de Decisão

Um formalismo amplamente utilizado na literatura para modelar problemas de decisões seqüenciais em ambientes não deterministas é o Processo Markoviano de Decisão (MDP - *Markovian Decision Process* [6]). A utilização dos MDPs deve-se principalmente ao seu formalismo matemático, inclusive já possuindo algoritmos para resolvê-los. Um MDP é definido por:

- um conjunto finito de ações possíveis $a \in \mathcal{A}$;
- um conjunto finito de estados do ambiente $s \in \mathcal{S}$;
- um processo markoviano discreto no tempo, modelado por probabilidades de transições $T(s'|s,a)$;
- uma função recompensa limitada $w(s,a) \in \mathbb{R}$; e
- a dinâmica do sistema é tal que, se o processo está no estado s_t no tempo t e a ação a_t é escolhida, então o próximo estado $s_{t+1} = s'$ ocorre de acordo com a probabilidade de transição $T(s'|s_t, a_t)$ e uma recompensa com valor $w(s_t, a_t)$ é recebida.

O agente deve encontrar uma política de ações $a_t = \pi^*(s_t)$ que maximize uma função valor V^π que representa as recompensas recebidas, isto é, $V^{\pi^*} = \max_{\pi \in \Pi} V^\pi$. Uma opção para a função valor é considerar simplesmente a soma de recompensas recebidas:

$$V^\pi = \mathrm{E}_{s_0 \sim P_0}[\sum_{t=0}^{N-1} w(s_t, a_t) | a_t = \pi(s_t)], \qquad (1)$$

onde N é o horizonte (número de passos) no qual a política será executada, e P_0 é uma distribuição de probabilidade para o estado inicial, isto é, $P_0(s_0 = s) \forall s \in \mathcal{S}$.

Um usuário pode programar um agente baseado em MDP definindo uma função recompensa apropriada às suas preferências. Embora o comportamento do agente possa ser guiado simplesmente pela definição da função recompensa, esta definição não está ao alcance de um usuário leigo, principalmente se suas preferências estabelecem compromissos entre vários objetivos: chegar a um estado alvo, não bater nas paredes, coletar objetos, etc. No entanto, representações mais compactas para a função recompensa podem ser utilizadas para facilitar a especificação da mesma.

Considere um conjunto de atributos Ξ com elementos $1, 2, \ldots, k$, e um mapeamento $\boldsymbol{\phi}$ de cada par (s,a) para um vetor de k valores de atributos observados, isto é, $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^k$ [7]. Pode-se considerar recompensas associadas linearmente aos valores de cada atributo, isto é, para cada atributo i existe uma recompensa correspondente w_i, gerando o vetor recompensa $\mathbf{w} = (w_i)_k$. Então, a função recompensa é definida por $w(s,a) = \langle \mathbf{w}, \boldsymbol{\phi}(s,a) \rangle$.

Uma política π também define um vetor médio de atributos

$$\boldsymbol{\mu}^\pi = \mathrm{E}_{s_0 \sim P_0}[\sum_{t=0}^{N-1} \boldsymbol{\phi}(s_t, a_t) | a_t = \pi(s_t)],$$

tal que se pode escrever a função valor como $V^\pi = \langle \mathbf{w}, \boldsymbol{\mu}^\pi \rangle$.

Dada o mapeamento $\boldsymbol{\phi}$, tudo que precisa ser especificado agora para estabelecer a função recompensa e programar o agente é o vetor $\mathbf{w}$ que estabelece o compromisso entre os diversos atributos. Note que, embora tal tarefa seja mais simples do ponto de vista do usuário, ainda é difícil especificar valores apropriados que representem as suas preferências. Na próxima seção será colocado como o processo de Extração de Preferências auxilia no processo de decisão quando o usuário não pode colocar de forma precisa suas preferências.

III. Tomando Decisões com Conhecimento Limitado

Nem sempre se pode obter a informação necessária para descrever integralmente as preferências do usuário. Suponha que o usuário possa impor algumas restrições como: o atributo i é mais importante que o atributo j, ou o atributo m é desejável, enquanto o atributo n deve ser evitado. Embora essas restrições possam limitar as políticas candidatas a otimalidade, elas não garantem que uma decisão ótima possa ser tomada.

Considere um conjunto W de possíveis vetores recompensa que atendem as restrições conhecidas, e um conjunto com todas políticas possíveis Π, pode-se optar pela política π^*_W de tal forma a minimizar o arrependimento (*regret*) de ter escolhido π^*_W no pior caso do conjunto W. Se o vetor recompensa $\mathbf{w}$, que modela as preferências do usuário, fosse conhecido, poder-se-ia definir o arrependimento de escolher uma política π como [8], [4]:

$$\mathrm{Regret}(\pi, \mathbf{w}) = \max_{\pi'} \langle \mathbf{w}, \boldsymbol{\mu}^{\pi^*} - \boldsymbol{\mu}^\pi \rangle.$$

Então, define-se π^*_W como sendo a política que causa o menor arrependimento quando considerando o pior caso dentre os vetores recompensas candidatos W, ou seja,

$$\pi^* = \arg \min_{\pi \in \Pi} \max_{\mathbf{w} \in W} \mathrm{Regret}(\pi, \mathbf{w}).$$

Essa definição representa uma tomada de decisão conservadora. Se W possui apenas um elemento, o arrependimento mínimo torna-se 0. Essa medida pode ser utilizada também como uma medida da acurácia do conhecimento sobre as preferências do usuário, uma vez que, quanto menor o arrependimento, maior é a garantia da precisão na tomada de decisão.

Como perguntar ao usuário diretamente sobre um vetor recompensa (ou uma função recompensa) que represente suas preferências é improdutivo, o processo de extração de preferências consiste em questionar o usuário com perguntas mais simples, mas cuja respostas resultam em mais restrições ao conjunto W. Algumas das estratégias baseiam-se diretamente nas opções de políticas disponíveis, por exemplo, garantindo que o arrependimento é sempre menor a cada questão. Portanto, é importante obter um método enumerativo e mínimo para tais políticas.

IV. Políticas e Vetores de Atributos Esperados

Um resultado importante na teoria de MDPs é que, dado um MDP $[\mathcal{S}, \mathcal{A}, T(s'|s,a),\ w(s,a)]$, se o horizonte de execução é finito, existe uma política ótima $\pi^*(s,t)$ determinista que o soluciona. Note que o estado deve ser estendido com o tempo de execução. Além disso, a política ótima pode ser definida recursivamente baseada na função valor $V^*(s,t)$ que mapeia a recompensa acumulada esperada a partir de s no tempo t até o tempo limite N de um período quando a política ótima é executada, isto é, $V^*(s,t) = \max_{\pi\in\Pi} \mathrm{E}[\sum_{i=t}^{N-1} w(s_i, \pi(s_i))|s_t = s]$. Define-se então:

$$V^*(s,t) = \begin{cases} 0, & \text{se } t \geq N \\ \max_{a\in\mathcal{A}}[w(s,a)+ \\ +\sum_{s'\in\mathcal{S}} T(s'|s,a)V^*(s',t+1)], & \text{se } t < N \end{cases}$$

e

$$\pi^*(s,t) = \arg\max_{a\in\mathcal{A}} w(s,a) + \sum_{s'\in\mathcal{S}} T(s'|s,a)V^*(s',t+1)\}.$$

O vetor de atributos esperados $\boldsymbol{\mu}^\pi$ de uma política determinista π pode ser calculado recursivamente pela equação:

$$\boldsymbol{\mu}^\pi(s,t) = \\ = \begin{cases} [0\ 0\ \cdots\ 0], & \text{se } t \geq N \\ \boldsymbol{\phi}(s,\pi(s))+ \\ +\sum_{s'\in\mathcal{S}} T(s'|s,\pi(s))\boldsymbol{\mu}^\pi(s',t+1), & \text{se } t < N \end{cases},$$

onde $\boldsymbol{\mu}^\pi(s,t)$ mapeia o vetor de atributos esperados a partir de s no tempo t até o tempo limite N de um período e $\boldsymbol{\mu}^\pi = \sum_{s\in\mathcal{S}} \Pr(s_0 = s)\boldsymbol{\mu}^\pi(s,0)$.

V. Propriedades do Espaço de Vetores de Atributos Esperados

A vantagem de trabalhar no espaço de vetores de atributos é que se pode definir um espaço métrico com a propriedade de que políticas vizinhas nesse espaço possuem valores próximos quando avaliadas por $\mathbf{w}$. Uma possível distância nesse espaço é simplesmente a distância euclidiana entre dois vetores de atributos. O espaço de atributos pode ser mais compacto, no sentido de que ele apresenta menor dimensão, isto é, enquanto um comportamento apresenta em sua dimensão a duração do período observado e como possíveis valores o espaço de estados $\mathcal{S}$; o espaço de atributos depende apenas do conjunto de atributos escolhido para estruturar as preferências do usuário.

Pode-se provar outra propriedade importante no espaço de vetores de atributos. Ao utilizar políticas estocásticas, o espaço de atributos permite valores contínuos e é um espaço convexo, no qual algoritmos de maximização baseados em gradientes podem ser implementados. Seja o conjunto de políticas estocásticas Π_{Est}, define-se o conjunto de todos os vetores de atributos esperados $\mathcal{M} = \{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{Est}}\}$. O fato do conjunto $\mathcal{M}$ ser convexo implica que, dado dois vetores $\boldsymbol{\mu}', \boldsymbol{\mu}'' \in \mathcal{M}$ e $\alpha \in [0,1]$, então $\boldsymbol{\mu}^\alpha \in \mathcal{M}$, onde $\boldsymbol{\mu}^\alpha = \alpha\boldsymbol{\mu}' + (1-\alpha)\boldsymbol{\mu}''$. Considere duas políticas π' e π'' tal que seus respectivos vetores de atributos esperados saõ $\boldsymbol{\mu}'$ e $\boldsymbol{\mu}''$, pode-se definir uma política π^α que resulta no vetor de atributos esperados $\boldsymbol{\mu}^\alpha$. A política π^α é definida para todo $a \in \mathcal{A}$ e todo $s \in \mathcal{S}$ por

$$\Pr(a|s,t,\pi^\alpha) = \begin{array}{l} \frac{\alpha \Pr(s|t,\pi')\Pr(a|s,t,\pi')}{\alpha \Pr(s|t,\pi')+(1-\alpha)\Pr(s|t,\pi'')} \\ +\frac{(1-\alpha)\Pr(s|t,\pi'')\Pr(a|s,t,\pi'')}{\alpha \Pr(s|t,\pi')+(1-\alpha)\Pr(s|t,\pi'')}. \end{array} \quad (2)$$

Então, dado um vetor de atributos $\boldsymbol{\mu}^\alpha$ convexo a quaisquer dois vetores de atributos esperados, tais que as políticas que os gerem são conhecidas, é possível construir uma política π^α que gera um vetor de atributos esperados $\boldsymbol{\mu}^\alpha$. O caso mais genérico, isto é, a política que gere um vetor de atributos esperados convexo ao conjunto de vetores de atributos esperados das políticas deterministas ($\{\boldsymbol{\mu}^\pi \mid \pi \in \Pi_{\text{Det}}\}$), também pode ser definido de modo equivalente. Dessa forma, o envoltório convexo $\mathcal{M} = \text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{Det}}\})$ determina um poliedro de vetores de atributos esperados que podem ser obtidos através de uma política estocástica apropriada.

O poliedro $\mathcal{M}$ é equivalente ao conjunto de todos os vetores de atributos esperados $\boldsymbol{\mu}^\pi$ gerados a partir das políticas estocásticas $\pi \in \Pi_{\text{Est}}$, pois se isso não fosse verdade, existiria uma função recompensa para qual não existiria uma política ótima em Π_{Det}. Na próxima seção será definida uma propriedade que permite a construção de um algoritmo que encontra os vértices do poliedro $\mathcal{M}$.

VI. Gerando um Conjunto de Políticas não Dominadas

O fato dos vetores de atributos esperados de todas as políticas estacionárias, sejam elas deterministas ou estocásticas, estarem limitados ao envoltório convexo $\mathcal{M} = \text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{Det}}\})$ das políticas deterministas Π_{Det} permite descrever os possíveis vetores de atributos esperados de forma compacta. Essa representação compacta é feita pelos vértices que representam tal poliedro e são vetores de atributos esperados associados a políticas não dominadas Π_{NonDom}, isto é, $\text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{NonDom}}\}) = \text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{Det}}\}) = \text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{Est}}\})$. Nesta seção, apresenta-se uma propriedade que permite gerar tais vértices, sem a necessidade de gerar todos vetores de atributos esperados que pertencem internamente ao envoltório convexo, isto é, não necessariamente todas políticas Π_{Det} precisam ser analisadas.

Os pontos extremos para cada um dos atributos necessariamente são vértices, isto é, os vetores de atributos esperados obtidos quando se considera MDPs com vetores recompensas baseados em apenas um dos atributos. Para o ponto extremo positivo deve-se maximizar o atributo i utilizando o vetor recompensa $\mathbf{w} = [0\ 0\ \cdots\ w_{i-1} = 0\ w_i = 1\ w_{i+1} = 0\ \cdots\ 0\ 0]$ e obtém-se $\boldsymbol{\mu}_i^{\max}$. O mesmo deve ser feito para o atributo i minimizando o seu valor, isto é, com

$\mathbf{w} = [0\ 0\ \cdots\ w_{i-1} = 0\ w_i = -1\ w_{i+1} = 0\ \cdots\ 0\ 0]$ e obtém-se $\boldsymbol{\mu}_i^{\min}$.

Com os vetores de atributos obtidos, pode-se criar limites mínimos para o conjunto factível de interesse. Se for considerado o envoltório convexo sobre esses pontos, isto é, $\widehat{\mathcal{M}} = \text{co}(\{\boldsymbol{\mu}_i^{\max} | i \in \Xi\} \cup \{\boldsymbol{\mu}_i^{\min} | i \in \Xi\})$, tem-se que $\widehat{\mathcal{M}} \subseteq \mathcal{M}$. Apesar desse conjunto ser uma aproximação para o conjunto de interesse, nada indica que é a melhor aproximação, outras direções poderiam ter sido escolhidas para definir tal aproximação. No entanto, como ela forma uma base, ela pode ser utilizada para determinar valores máximos e mínimos que uma política pode obter consoante um vetor recompensa arbitrário.

Para determinar o conjunto completo $\mathcal{M} = \text{co}(\{\bar{\boldsymbol{\mu}}_\pi | \pi \in \Pi_{\text{Det}}\})$ pode-se inicialmente utilizar o conjunto $\widehat{\mathcal{M}}$ e acrescentar novos vértices ao mesmo até se obter $\mathcal{M}$. A estratégia é testar para todas as faces do poliedro $\widehat{\mathcal{M}}$ se elas também são faces reais do poliedro $\mathcal{M}$, caso não seja uma face real, encontra-se um ponto além dela e que pertença ao poliedro $\mathcal{M}$ para definir novas faces candidatas. Esse procedimento é assegurado pelo seguinte teorema.

Teorema 1 *Seja o vetor recompensa* $\mathbf{w}$ *e uma política ótima* $\pi^*_{\mathbf{w}}$ *com seu respectivo vetor de atributos esperados* $\boldsymbol{\mu}^{\pi^*_{\mathbf{w}}}$*. Ainda, considere o hiperplano* H *que contém o ponto* $\boldsymbol{\mu}^{\pi^*_{\mathbf{w}}}$ *e é perpendicular a* $\mathbf{w}$*. Então, os vetores de atributos esperados que estão além do hiperplano* H*, com respeito a direção* $\mathbf{w}$*, não pertencem ao conjunto* $\text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{Est}\})$.

Proof: Considere um vetor de atributos qualquer $\boldsymbol{\mu}$, a operação $\langle \mathbf{w}, \boldsymbol{\mu} \rangle$ avalia o vetor de atributos $\boldsymbol{\mu}$ segundo o vetor recompensa $\mathbf{w}$. Isso implica que $\langle \mathbf{w}, \boldsymbol{\mu}^\pi \rangle \leq \langle \mathbf{w}, \boldsymbol{\mu}^{\pi^*_{\mathbf{w}}} \rangle$ para todo $\pi \in \Pi$, pois $\pi^*_{\mathbf{w}}$ é uma política ótima segundo $\mathbf{w}$.

Por outro lado, o hiperplano H é definido pelo vetor $\mathbf{w}$ e o valor $\langle \mathbf{w}, \boldsymbol{\mu}^{\pi^*_{\mathbf{w}}} \rangle$, pois $\mathbf{w}$ é perpendicular a H e $\boldsymbol{\mu}^{\pi^*_{\mathbf{w}}}$ pertence ao plano. Logo, $\boldsymbol{\mu} \in H$ se e somente se $\langle \mathbf{w}, \boldsymbol{\mu} \rangle = \langle \mathbf{w}, \boldsymbol{\mu}^{\pi^*_{\mathbf{w}}} \rangle$ e $\boldsymbol{\mu}$ está além de H se e somente se $\langle \mathbf{w}, \boldsymbol{\mu} \rangle > \langle \mathbf{w}, \boldsymbol{\mu}^{\pi^*_{\mathbf{w}}} \rangle$.

Considere que exista um vetor de atributos esperado $\boldsymbol{\mu}'$ além de H e que $\boldsymbol{\mu}' \in \text{co}(\{\boldsymbol{\mu}^\pi | \pi \in \Pi_{\text{Est}}\})$, isto é, o vetor de atributos $\boldsymbol{\mu}'$ possui uma política, mesmo que estocástica, que o obtenha como vetor de atributos esperados. Então existe uma política melhor que $\pi^*_{\mathbf{w}}$, resultando em contradição. Logo, esse não pode ser o caso. ■

Como resultado, esse teorema permite propor uma regra para incrementar o conjunto auxiliar $\widehat{\mathcal{M}}$ até que o mesmo represente o conjunto de vetores de atributos esperados factíveis $\mathcal{M}$. Inicialmente se considera o poliedro $\widehat{\mathcal{M}} = \text{co}(\{\bar{\boldsymbol{\mu}}_i^{\max} | i \in \Xi\} \cup \{\bar{\boldsymbol{\mu}}_i^{\min} | i \in \Xi\})$ (figura 1a). Considere um hiperplano H' que limita o poliedro $\widehat{\mathcal{M}}$ e o respectivo vetor $\mathbf{w}_{H'}$ perpendicular ao plano H' e que aponta para fora do poliedro $\widehat{\mathcal{M}}$ (figura 1b e figura 1c). Encontre uma política ótima $\pi^*_{\mathbf{w}_{H'}}$ para $\mathbf{w}_{H'}$ e o respectivo vetor de atributos esperados $\bar{\boldsymbol{\mu}}^{\pi^*_{\mathbf{w}_{H'}}}$. Se $\bar{\boldsymbol{\mu}}^{\pi^*_{\mathbf{w}_{H'}}}$ está além de H', então $\bar{\boldsymbol{\mu}}^{\pi^*_{\mathbf{w}_{H'}}}$ pode ser considerado como mais um vértice do poliedro auxiliar $\widehat{\mathcal{M}}$ (figura 1b), caso contrário o hiperplano H' limita o conjunto $\mathcal{M}$ (figura 1c). Se todos os hiperplanos que limitam $\widehat{\mathcal{M}}$ também limitam o conjunto $\mathcal{M}$, então $\widehat{\mathcal{M}} = \mathcal{M}$. O fim desse processo é garantido, uma vez que a cardinalidade de Π_{Det} é finita.

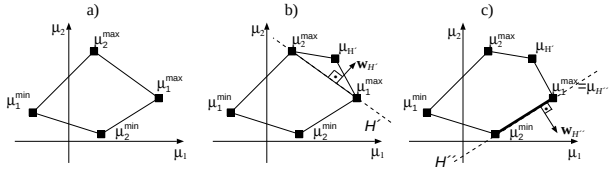

Figure 1. Gerando um conjunto de vetores de atributos esperados factíveis. a) Poliedro $\widehat{\mathcal{M}}$ inicial tendo como vértices os vetores que minimizam e maximizam cada um dos atributos isoladamente. b) Exemplo de um plano do poliedro $\widehat{\mathcal{M}}$ que não pertence ao poliedro $\mathcal{M}$. c) Exemplo de um plano do poliedro $\widehat{\mathcal{M}}$ que pertence ao poliedro $\mathcal{M}$.

VII. Experimentos

Os objetivos dos experimentos realizados é demonstrar o quão compacto a representação do conjunto de políticas Π_{Est} pode ser no espaço de vetores de atributos esperados. Tal conjunto é representado apenas pelos vértices do poliedro que forma o envoltório convexo dos vetores de atributos de todas políticas estocásticas. O algoritmo para encontrar tal conjunto é testado sob diferentes níveis de precisão, mostrando que a representação pode ser ainda mais compacta, quando quase-otimalidade é considerada.

A. Tarefa do Agente

A tarefa que se deseja programar no agente é a tarefa de se deslocar de um ponto inicial a um ponto final. Além de completar tal tarefa, o agente deve maximizar ou minimizar a ocorrência de alguns atributos: `distância` (μ_1), `tempo` (μ_2), `curva geral` (μ_3), `curva fechada` (μ_4) e `colisão` (μ_5). No entanto, o compromisso entre tais atributos deve satisfazer as preferências de um usuário.

Na figura 2 é exibido o ambiente considerado nos experimentos. O agente deve partir da posição inicial s_i e chegar na posição final s_f com no máximo 25 ações. Também são exibidas as 105 possíveis posições discretas assumidas pelo agente. Além da posição do agente, também é considerada uma discretização da direção do mesmo, considerando 8 possíveis valores (discretização de 45°). Considerando ainda a passagem do tempo, o robô pode então assumir 21000 estados.

O agente pode escolher, a cada iteração, uma dentre um conjunto de 12 ações: mover-se uma célula em uma das quatro direções cardeais, mover-se duas células em uma das quatro direções cardeais[1], mover-se uma célula em uma das quatro diagonais. Se a próxima posição determinada por uma ação não pode ser ocupada devido a presença

[1]Mesmo que exista um obstáculo entre a posição inicial e a posição final, o agente contorna tal obstáculo para chegar à posição desejada.

de obstáculos, então esta ação não consta como opção ao agente (ver exemplo na figura 2, onde, das 12 opções de ações, somente 8 estão disponíveis na situação mostrada). Devido a configuração do ambiente, a quantidade média de ações disponíveis para cada estado é aproximadamente 6,5. O espaço de estados e ações resultam então em um espaço de políticas com $21000^{6,5}$ opções de decisão, ou seja, da ordem de 10^{28} políticas.

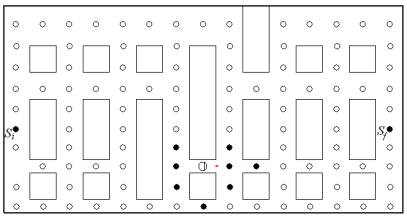

Figure 2. Ambiente utilizado nos experimentos para definir conjunto de vetores de atributos factíveis. O tamanho do ambiente é um retângulo com $30m \times 15m$ e obstáculos distribuídos no mesmo de forma estruturada. O robô deve utilizar como posições intermediárias as 105 posições distribuídas no ambiente. O robô inicializa na posição s_i e deve alcançar a posição s_f. Um robô localizado como na figura, em um passo de tempo, pode se mover para qualquer uma das 8 posições marcadas próximas ao robô.

B. Montagem Experimental

A tarefa de obter uma trajetória entre dois pontos foi simulada por um arcabouço próximo da realidade, o simulador *ARIA* [9]. Este simulador simula o robô *Pioneer*, que é um robô móvel não holonômico. Além dos sonares, o robô também possui um sensor de movimento, o hodômetro, que permite estimar a distância percorrida pelo robô.

Junto ao simulador é utilizada a plataforma *SAPHIRA*, que implementa um algoritmo de localização com base em localização markoviana e um algoritmo para controle e planejamento de trajetória baseado em gradientes com relação a posição desejada e os obstáculos no ambiente [9]. O algoritmo de localização permite minimizar os erros encontrados nas medidas feitas pelos sensores do robô (sonares e hodômetro), e, ao integrar as medidas acumuladas ao longo do tempo, faz com que a postura estimada do robô seja próxima da postura real (a nuvem de pontos vermelhos na figura 3 representa possíveis posições onde o robô pode estar). O algoritmo de planejamento permite realizar uma discretização mais grosseira do ambiente, fazendo com que a escolha de ações seja feita sempre em posições discretas a menos de um erro (o robô se posiciona no máximo a $30cm$ do ponto escolhido). Na figura 3, o caminho em vermelho é a trajetória planejada até o alvo pelo algoritmo com base em gradientes.

A observação dos 5 atributos considerados pelo agente foi implementada utilizando os sensores do robô. Após um evento discreto de tempo fixo Δt mede-se a variação na posição do robô Δd e a variação da direção do robô $\Delta\theta$. Os atributos `distância` e `tempo` medem respectivamente a distância percorrida pelo agente da postura

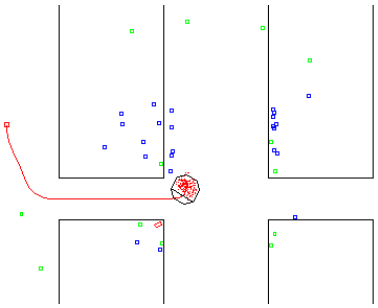

Figure 3. Localização e planejamento na plataforma *SAPHIRA*. A nuvem de pontos vermelhos representa possíveis posições do robô, às quais probabilidades de aderência são associadas. O caminho em vermelho representa a trajetória planejada pelo algoritmo baseado em gradiente para atingir a posição alvo.

inicial até a posição final ($\sum \Delta d$) e o tempo gasto para realizar esse percurso ($\sum \Delta t$). O atributo `curva geral` considera o acúmulo de variações da direção do robô ao longo do percurso ($\sum \Delta\theta$), enquanto o atributo `curva fechada` acumula apenas a variação da direção quando há pouco movimento de translação, isto é, mede curvas mais fechadas. O atributo `curva fechada` foi implementado como $\sum \max\{0, \Delta\theta - 40 * \Delta d\}^2$, com Δd medido em *metros* e $\Delta\theta$ medido em *graus*. O atributo `colisão` foi implementado analisando o estado dos motores do robô, quando os motores se encontram impedidos de rotacionar, o robô locomove-se em sentido contrário e uma colisão é contabilizada.

Por último, para que algoritmos de planejamento possam ser utilizados, experimentos preparatórios foram realizados para determinar as probabilidades de transições $T(s'|s,a)$ entre os estados para cada ação e ainda a função de observação $\phi(s,a)$. Essa informação foi levantada durante a repetição sistemática de ao menos 30 vezes da execução de todas as ações em cada uma das possíveis situações.

C. Resultados

Utilizando as funções $T(s'|s,a)$ e $\phi(s,a)$, é possível calcular a política ótima e o respectivo vetor de atributos esperados para um vetor recompensa $\mathbf{w}$ arbitrário. Calculou-se então os vetores de atributos esperados $\boldsymbol{\mu}_i^{\max}$ e $\boldsymbol{\mu}_i^{\min}$ referentes às respectivas políticas ótimas para os vetores recompensas $\mathbf{w}_i^{\max}$ e $\mathbf{w}_i^{\min}$ respectivamente para $i = 1, 2, \ldots, 5$ (ver seção VI).

Devido ao grande número de políticas disponíveis, o número de políticas não dominadas ainda pode ser muito alto, no entanto muitas dessas políticas podem trazer pouca melhoria quando confrontadas com outras políticas, mesmo que se utilize uma avaliação arbitrária. Dessa forma, algumas políticas não dominadas podem ser descartadas, sem que ocorra grande perda ao escolher uma política ótima apenas

[2]O coeficiente 40 foi ajustado para que apenas curvas com raio menor que 1,5 metros sejam contabilizadas.

entre as políticas restantes. Então, considerou-se o algoritmo para encontrar os vetores atributos esperados factíveis $\widehat{\mathcal{M}}^{\epsilon}$ sob a limitação de um erro ao testar se um hiperplano pertence ou não pertence ao poliedro que determina $\widehat{\mathcal{M}}^{\epsilon}$. Seja $\boldsymbol{\mu} \in H$; $\mathbf{w}_H$, o vetor recompensa perpendicular ao hiperplano H; $\boldsymbol{\mu}^{*\pi_{\mathbf{w}_H}}$, o vetor de atributos esperados da política ótima sob o vetor recompensa $\mathbf{w}_H$; e ϵ um erro arbitrário; então considera-se $H \in \widehat{\mathcal{M}}^{\epsilon}$ se $\langle \mathbf{w}_H, \boldsymbol{\mu} \rangle + \epsilon \geq \langle \mathbf{w}_H, \boldsymbol{\mu}^{*\pi_{\mathbf{w}_H}} \rangle$.

Nos experimentos foram considerados quatro valores de erros ϵ. Na tabela I os resultados obtidos são exibidos: para cada ϵ é exibida a quantidade de vetores de atributos esperados não dominados utilizados para representar $\widehat{\mathcal{M}}^{\epsilon}$, a quantidade de hiperplanos utilizados para representar $\widehat{\mathcal{M}}^{\epsilon}$, o hipervolume normalizado que o conjunto $\widehat{\mathcal{M}}^{\epsilon}$ cobre e a porcentagem de cobertura com relação ao poliedro de vetores atributos esperados factíveis. Para $\epsilon = 0,0001$, o algoritmo foi finalizado antes da convergência, isto é, a verificação de todos os hiperplanos, no entanto, o volume alcançado foi considerado o máximo de forma aproximada dentro da precisão adotada na tabela I. Quando comparado ao volume do conjunto inicial, mesmo para $\epsilon = 0,1$, o volume obtido é da ordem de dezenas de vezes maior que o volume do conjunto inicial. No entanto, para $\epsilon = 0,0001$ ocorre pouca variação no volume do conjunto de vetores de atributos esperados com relação a $\epsilon = 0,001$, mas a quantidade de vértices e a quantidade de hiperplanos é pelo menos três vezes maior.

Table I
CALCULANDO VETORES DE ATRIBUTOS ESPERADOS NÃO DOMINADOS.

ϵ	vértices	hiperplanos	volume ($\times 10^{-3}$)	% de cobertura
(início)	10	40	0,0015	0,13
0,1	15	90	0,1338	11,17
0,01	215	4862	0,9979	84,91
0,001	1214	30524	1,1753	98,08
0,0001	>4048	>101008	≈1,1983	≈100,00

Embora o espaço de políticas possíveis possa ser muito grande para um determinado ambiente, a ordem de grandeza do número de políticas que realmente devem ser consideradas pelo agente, isto é, as políticas não dominadas, pode ser muito menor. No entanto, o conjunto de políticas não dominadas ainda apresenta uma alta cardinalidade. No entanto, se considerações sobre quase-otimalidade são feitas, o conjunto de políticas a serem consideradas pode ser ainda mais reduzido, sem trazer piora significativa para o desempenho do agente.

VIII. CONCLUSÃO

Neste trabalho foi apresentado um algoritmo para especificar todas políticas não dominadas de um determinado problema de tomada de decisões seqüenciais. A definição de tal conjunto é essencial para várias estratégias de extração de preferências, já que ele permite especificar quais são as opções de decisões que merecem ser analisadas de fato.

O algoritmo permite calcular tal conjunto de forma aproximada, permitindo a escolha de uma precisão arbitrária. A possibilidade de tal escolha é importante, pois na maioria dos problemas, definir o conjunto completo de políticas não dominadas é inviável, já que este número pode crescer exponencialmente com a quantidade de estados do problema.

Por outro lado, por meio de um experimento simples mas realista, mostrou-se que um conjunto de políticas que cubra o espaço de resultados possíveis de forma aproximada pode ser muito menor que o espaço completo.

AGRADECIMENTOS

Este trabalho foi conduzido sob os projetos Ob-SLAM (CNPq proc. 475690/2008-7) e LogProb (FAPESP proc. 2008/03995-5). Valdinei F. Silva também agradece a FAPESP (proc. 02/13678-0 e proc. 09/14650-1) e CAPES (proc. BEX-3388/04-2). Anna H. R. Costa também agradece ao projeto LMBot (CNPq proc. 305512/2008-0).

REFERENCES

[1] B. M. Dennis and C. G. Healey, "A survey of preference elicitation," Knowledge Discovery Lab, Department of Computer Science, North Carolina State University, Tech. Rep., 2003.

[2] D. Braziunas, "Computational approaches to preference elicitation," Department of Computer Science, University of Toronto, Tech. Rep., 2006.

[3] C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans, "Constraint-based optimization and utility elicitation using the minimax decision criterion," *Artificial Intelligence*, vol. 170, no. 8, pp. 686–713, 2006.

[4] ——, "Regret-based utility elicitation in constraint-based decision problems," in *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland: Professional Book Center, 2005, pp. 929–934.

[5] S. Buffett and M. W. Fleming, "Persistently effective query selection in preference elicitation," in *IAT '07: Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 491–497.

[6] S. M. Ross, *Applied probability models with optimization applications*. San Francisco: Holden-Day, 1970.

[7] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 663–670.

[8] T. Wang and C. Boutilier, "Incremental utility elicitation with the minimax regret decision criterion," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*. Acapulco: Morgan Kaufmann, 2003, pp. 309–316.

[9] *Saphira's Manual*, ActivMedia Robotics, Menlo Park, CA, 2001, version 8.0a.