Exact Algorithm to Evaluate Stationary Policies for CVaR MDP

Denis B. Pais[1], **Karina V. Delgado**[1], **Valdinei Freire**[1]

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (EACH - USP)
São Paulo – SP – Brasil

{denis.pais,kvd,valdinei.freire}@usp.br

Abstract. *Markov Decision Processes (MDPs) are widely used to model sequential-decision problems. The most widely used performance criterion in MDPs is the minimization of the total expected cost. However, this approach does not take into account variations around the mean, and very different results are evaluated equally; MDPs that deal with this kind of problem are called risk sensitive MDPs. A special type of risk-sensitive MDP is CVaR MDP, which includes the CVaR (Conditional-Value-at-Risk) metric commonly used in finance. One algorithm that finds the optimal policy for CVaR MDPs is the Linear Interpolation Value Iteration algorithm called CVaRVILI. To obtain an approximated optimal policy, the CVaRVILI algorithm solves linear programming problems several times, and CVaRVILI algorithm is computational expensive. In this work, we propose the PECVaR algorithm that evaluates stationary policies for CVaR MDPs under constant cost and does not need to solve linear programming problems. Although PECVaR algorithm cannot be used to improve a policy, we hypothesized that the CVaR value of an arbitrary policy is a better initialization to CVaRVILI algorithm than the expected value of the same policy. We evaluate empirically such a hypothesis and the results show that using PECVaR initialization decreases the CVaRVILI algorithm convergence time in most cases.*

Resumo. *Processos de decisão Markovianos (*Markov Decision Processes – *MDPs) são amplamente utilizados para resolver problemas de tomada de decisão sequencial. O critério de desempenho mais utilizado em MDPs é a minimização do custo total esperado. Porém, esta abordagem não leva em consideração variações em torno da média, e resultados diferentes são avaliados igualmente; MDPs que lidam com esse tipo de problema são chamados de MDPs sensíveis a risco. Um tipo especial de MDP sensível a risco é o CVaR MDP, que inclui a métrica CVaR (*Conditional-Value-at-Risk*) comumente utilizada na área financeira. Um algoritmo que encontra a política ótima para CVaR MDPs é o algoritmo de Iteração de Valor com Interpolação Linear chamado CVaRVILI. Para obter uma solução ótima aproximada, o algoritmo CVaRVILI resolve problemas de programação linear várias vezes, e o algoritmo CVaRVILI tem um alto custo computacional. Neste trabalho é proposto o algoritmo PECVaR que avalia uma política estacionária para CVaR MDPs de custo constante e não precisa resolver problemas de programação linear. Embora o algoritmo PECVaR não possa ser utilizado para melhorar uma política, considera-se a hipótese de que o valor CVaR de uma política arbitrária é melhor como inicialização do algoritmo CVaRVILI do que o valor esperado da*

mesma política. Tal hipótese é avaliada empiricamente e os resultados mostram que utilizando PECVaR para inicializar o algoritmo CVaRVILI pode diminuir o tempo de convergência do CVaRVILI na maioria dos casos.

1. Introdução

A cada dia milhares de decisões são tomadas por agentes humanos e não humanos, decisões com consequências imediatas e a longo prazo. Geralmente decisões não podem ser feitas de forma isolada, pois as decisões tomadas agora podem afetar as decisões futuras. Quando essa relação entre o resultado presente e futuro não é considerada, temos uma grande chance de não atingirmos um bom desempenho. Tais problemas motivam o desenvolvimento de diferentes técnicas para melhorar a modelagem de processos de tomada de decisão.

Um processo de decisão de Markov (*Markov Decision Process*–MDP) é um modelo matemático usado para modelar problemas de tomada de decisões sequenciais nos quais as transições entre estados são probabilísticas, é possível observar em que estado o processo se encontra e é possível interferir no processo periodicamente (em épocas de decisão) executando ações Puterman. (1994). Cada ação tem um custo, que depende do estado em que o processo se encontra. A resolução de MDPs envolve a minimização de uma determinada função de desempenho. A função mais utilizada é o custo total descontado esperado, que é neutra em termos de risco. Esta abordagem, embora muito popular, natural e atraente de um ponto de vista computacional, não leva em consideração a variabilidade do custo, nem sua sensibilidade aos erros de modelagem, o que pode afetar significativamente o desempenho geral do processo Chow, Tamar, Mannor, e Pavone (2015).

Em muitas situações é preciso garantir com certo grau de certeza que será obtido um determinado resultado. Por exemplo, em um sistema de navegação autônomo, um agente tentará minimizar o comprimento esperado do seu caminho, e para isso ele pode provavelmente viajar perto de obstáculos na esperança de minimizar a distância. Entretanto esse mesmo agente também viajará perto de outros agentes e até mesmo de seres humanos, e uma falha ou desvio do caminho planejado pode resultar em uma colisão, ou um grave acidente causando uma perda irreversível (por exemplo, a morte de uma pessoa). Conseguimos lidar com esses problemas adicionando uma medida de risco, tornando o MDP sensível a risco.

Diversas medidas para mensuração de risco financeiro têm sido constantemente estudadas e aplicadas em diversos setores. Tais medidas são comumente empregadas em modelos de otimização estocástica aplicados para problemas do mercado financeiro e também da engenharia em geral. Entre essas medidas estão: variância, *Value-at-Risk* (VAR) e *Conditional-Value-at-Risk* (CVaR).

CVaR é considerada a principal e mais promissora métrica de risco. O CVaR é definida como a perda esperada condicionada ao fato de se estar no ponto (100 - c%) da cauda esquerda da distribuição. Diversos trabalhos sobre MDPs sensíveis a risco que utilizam esse critério foram propostos, entre eles Chow et al. (2015), Iyengar e Ma (2013) e Carpin, Chow, e Pavone (2016). Esse MDP que usa CVaR como métrica de risco é chamado de CVaR MDP. Um algoritmo que encontra a política ótima para CVaR MDPs é o algoritmo de Iteração de Valor com Interpolação Linear, chamado CVaRVILI Chow et

al. (2015). Porém, esse algoritmo é muito custoso pois precisa resolver vários problemas de programação linear. Neste trabalho, é proposto um algoritmo que avalia de maneira exata uma política estacionária para CVaR MDPs de custo constante que não precisa resolver problemas de programação linear e que tem um custo computacional similar ao de um algoritmo de iteração de valor para MDPs neutros ao risco.

Esse texto está organizado da seguinte forma. A seção 2 apresenta os fundamentos teóricos sobre Processsos de Decisão Markovianos. A seção 3 apresenta as métricas de risco VaR e CVaR, além das definições sobre CVaR MDP. Na seção 4 é descrito o algoritmo exato de avaliação de uma política estacionária para CVaR MDPs proposto neste trabalho. Na seção 5, o algoritmo proposto é usado para inicializar o algoritmo CVaR-VILI. Por fim, a seção 6 apresenta as conclusões.

2. Processos de Decisão Markovianos

Um processo de decisão Markoviano (Markov Decision Process - MDP) Puterman. (1994) é uma tupla $M = (S, A, P, C, \gamma, s_0)$, em que: S é um conjunto discreto e finito de estados completamente observáveis que modelam o mundo; A é um conjunto finito de ações; $P(\cdot|s, a)$ é a função probabilística de transição que descreve os efeitos da execução de uma ação $a \in A$ em um estado $s \in S$ resultando em um estado $s' \in S$; $C(s, a)$ é a função custo de executar uma ação $a \in A$ em um estado $s \in S$; e γ é o fator de desconto, sendo $0 \leq \gamma < 1$.

O agente executa as ações em passos discretos no tempo. A cada ação executada, o estado do sistema é alterado segundo a função de transição P, sendo que a execução de uma ação em um estado tem um custo.

A tomada de decisão é realizada durante um horizonte. O horizonte é o número de passos (ou épocas de decisão) que o agente tem para agir. Assim, MDPs podem ser classificados por seu horizonte em: (i) MDP com *horizonte finito* que tem um número de passos fixo a se tomar; (ii) MDP com *horizonte infinito* em que os passos são tomados repetidamente, sem a possibilidade de parada; e (iii) MDP com *horizonte indeterminado* que termina assim que o processo chega em algum estado meta.

A solução de um MDP é uma política. Considerando a sua relação com os passos (ou épocas de decisão), uma politica pode ser classificada como: (i) *estacionária*, se a ação recomendada no estado s independe do número de passos de decisão; e (ii) *não-estacionária*, caso contrário. Uma politica pode também ser classificada como: (i) *determinística*, quando cada estado $s \in S$ é sempre mapeado em uma única ação; e (ii) *estocástica*, quando um estado é mapeado em um conjunto de ações, sendo que cada ação tem uma probabilidade de ser escolhida. O foco deste trabalho é políticas determinísticas e MDPs de horizonte indeterminado. Este tipo de MDPs inclui um conjunto de estados meta G.

A solução de um MDP é uma política estacionária π: $S \rightarrow A$ que especifica a ação $a = \pi(s)$ a ser escolhida em cada estado s. O valor da política π começando no estado s e executando π, é denotado por $V_\pi(s)$ e é definido como a soma esperada dos custos descontados, em que $s_0 = s$, isto é:

$$V_\pi(s) = E_\pi\Big[\sum_{k=0}^{\infty} \gamma^k c_k | s_0 = s\Big] \tag{1}$$

Uma política gulosa π_V com respeito a alguma função valor $V : S \to \mathbb{R}$ é definida como uma política que escolhe uma ação em cada estado s e que minimiza o valor esperado com respeito a V, conforme definida a seguir Puterman. (1994):

$$\pi_V(s) = \arg\min_{a \in A}\{Q(s,a)\} = \arg\min_{a \in A}\left\{C(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)V(s')\right\} \quad (2)$$

em que Q(s, a) é o valor do estado s aplicando a ação a.

Dentre todas as possíveis políticas para um MDP, deseja-se encontrar uma política ótima π^* que minimize o custo total esperado descontado. A função valor ótima, representada por V^*, é a função valor associada com qualquer política ótima. Assim, para um agente que deseja minimizar seu custo total esperado descontado, V^* satisfaz a seguinte igualdade de ponto fixo Bellman (1957):

$$V^*(s) = \min_{a \in A}\{C(s,a) + \gamma \sum_{s' \in S} P(s'|s,a)V^*(s')\}. \quad (3)$$

A política ótima pode ser obtida aplicando $argmin$ na equação 3, no lugar de min. MDPs que minimizam o custo total esperado descontado são considerados neutros a risco pois não levam em consideração flutuações em torno da média. Os algoritmos clássicos para resolver MDPs neutros ao risco são *Iteração de Valor*, *Iteração de Política* e a formulação usando programação linear.

3. MDP e risco

Nesta seção são descritas as métricas VaR (*Value at Risk*) e CVaR (*Conditional-Value-at-Risk*) muito utilizadas para gestão de portfólio de ativos financeiros (chamados também de ações). Além disso, são descritos MDPs que usam a métrica CVaR na função objetivo.

3.1. Métricas VaR e CVaR

VaR mede a pior perda esperada ao longo de determinado intervalo de tempo sob condições normais e dentro de determinado nível de confiança α. VaR com nível de confiança $\alpha \in (0, 1)$ é definido como o quantil $1 - \alpha$ de Z, i.e.:

$$VaR_\alpha(Z) = \min\{z|F(z) \geq \alpha\}, \quad (4)$$

em que Z é uma variável aleatória e neste trabalho é interpretado como custo acumulado.

Uma medida alternativa é a função CVaR. Essa medida indica de forma mais adequada o potencial de perdas que ultrapassam o intervalo de confiança, definido ao se calcular a média das perdas que excedem o valor do VaR. Além disso, CVaR é considerada uma medida coerente de risco, não precisa de um distribuição normalizada para Z e apresenta uma maior estabilidade pois a flutuação sobre perturbações é menor quando comparada com VaR. CVaR pode ser definida, com nível de confiança $\alpha \in (0, 1)$ da seguinte forma Uryasev e Rockafellar (2001):

$$CVaR_\alpha(Z) = \min_{w \in \mathbb{R}}\{w + \frac{1}{\alpha}\mathbb{E}[(Z - w)^+]\}, \quad (5)$$

em que $(x)^+ = \max(x, 0)$, representa a parte positiva de x, Z é uma variável aleatória e w representa a variável de decisão que, no ponto ótimo, atinge o valor do VaR.

3.2. CVaR MDP

A função objetivo utilizada em MDPs, que é neutra a risco, pode ser substituída por alguma medida de risco como VaR ou CVaR. Neste trabalho, examina-se MDPs sensíveis a risco com a função objetivo CVaR, referido como CVaR MDP Chow et al. (2015).

Existe uma representação dual de CVaR que ajuda a viabilizar sua utilização em problemas de tomada de decisão sequencial, definida como Pflug e Pichler (2016):

$$CVaR_{\alpha}(Z) = \max_{\xi \in \mathcal{U}_{CVAR}(\alpha,\mathbb{P})} \mathbb{E}_{\xi}[Z], \tag{6}$$

em que Z é uma variável aleatória, que no nosso caso representa o custo acumulado, $\mathbb{E}_{\xi}[Z]$ é a esperança ξ-ponderada de Z, U_{CVAR} é chamado de envelope de risco e é representado da seguinte forma Chow et al. (2015):

$$\mathcal{U}_{CVaR}(\alpha,\mathbb{P}) = \left\{ \xi : \xi(\omega) \in [0, \frac{1}{\alpha}], \int_{\omega \in \Omega} \xi(\omega)\mathbb{P}(\omega)d\omega = 1 \right\}, \tag{7}$$

em que $\mathbb{P}$ é uma medida de probabilidade e $\omega \in \Omega$, sendo Ω o espaço amostral.

O envelope de risco pode ser visto como um conjunto de medidas de probabilidade que fornecem alternativas para a medida de probabilidade $\mathbb{P}$. Nesse caso, a medida de desvio correspondente $\mathcal{U}_{CVaR}(\alpha,\mathbb{P})$ estima a diferença do que o agente pode esperar sob $\mathbb{P}$ e sob a pior distribuição de probabilidade. Assim, CVaR de Z também pode ser interpretada como a *esperança do pior caso de Z* sobe a distribuição perturbada $\xi\mathbb{P}$ Chow et al. (2015).

Baseados nessa representação dual de CVaR e utilizando outros resultados de Pflug e Pichler (2016), em Chow et al. (2015) é apresentada uma solução baseada em programação dinâmica (DP) para o problema CVaR MDP. Nessa abordagem é utilizado um MDP com estados estendidos, ou seja o espaço de estados S foi incrementado com Y = (0, 1], que representam o nível de confiança α. Também são estabelecidas algumas propriedades importantes dessa formulação de DP, que permitem derivar um algoritmo eficiente e aproximado. O Teorema 1 é utilizado para derivar a formulação usada no casamento entre MDP e CVaR Pflug e Pichler (2016).

Teorema 1. *(Decomposição do CVaR) Para qualquer t ≥ 0, seja Z = (Z_{t+1}, Z_{t+2},...) a sequência de custos a partir do tempo t+1 em diante. O CVaR considerando a política μ, obedece a seguinte decomposição:*

$$CVaR_{\alpha}\Big(Z|h_t,\mu\Big) = \max_{\xi \in \mathcal{U}_{CVAR}(\alpha,P(.|s_t,a_t))} \mathbb{E}\left[\xi(s_{t+1}) \cdot CVaR_{\alpha\xi(s_{t+1})}\Big(Z|h_{t+1},\mu\Big) \middle| h_t,\mu \right], \tag{8}$$

em que h_t é o histórico até o tempo t, a_t é a ação induzida pela política $\mu_t(h_t)$, e a esperança é em relação ao estado s_{t+1}.

Note que a decomposição recursiva descrita no Teorema 1, apresenta diferentes termos CVaR do lado esquerdo e direito, com diferentes níveis de confiança (α no lado esquerdo e $\alpha\xi(s_{t+1})$ no lado direito). Dessa forma, o espaço de estados S precisa ser aumentando com o nível de confiança, que é contínuo, representado por Y = (0, 1] Chow et al. (2015). Logo, a função valor V está em função de $s \in S$ e $y \in Y$ e é definida como:

$$V(s,y) = \min_{\mu \in \Pi_H} CVaR_{y}\Big(\lim_{T\to\infty} C_{0,T}|s_0 = s,\mu\Big), \tag{9}$$

em que $\mu = \{\mu_0, \mu_1, ...\}$ é a sequência de políticas com ações $a_t = \mu_t(h_t)$ para $t \in \{0, 1, ...\}$ e $C_{0,T} = \sum_{t=0}^{T} \gamma^t Z_t$, isto é, o custo total descontado até o tempo T.

Na programação dinâmica é muito conveniente usar operadores que são definidos no espaço da função valor. O teorema da decomposição do CVaR (Teorema 1) conduziu à criação do operador de Bellman $T : S \times Y \to S \times Y$ para CVaR, definido a seguir:

$$T[V](s,y) = \min_{a \in A}\{Q(s,y,a)\} = \min_{a \in A}\Big\{C(s,a) + \gamma \max_{\xi \in \mathcal{U}_{CVAR}(y,P(.|s,a))} \sum_{s' \in S} \xi(s') V\Big(s', y\xi(s')\Big) P(s'|s,a)\Big\}, \tag{10}$$

em que o envelope de risco $\mathcal{U}_{CVAR}$ é definido pela equação 7. Na equação 10 é escolhida a melhor ação e é feita uma maximização contínua da expressão considerando o envelope de risco. Essa equação tem duas propriedades fundamentais para a tratabilidade dos problemas que são a contração e concavidade em y. Porém, esse operador de Bellman apresenta duas dificuldades: (i) nos estados aumentados, Y é contínua; e (ii) aplicar T envolve realizar a maximização sobre ξ.

Em Chow et al. (2015) foi proposto um algoritmo chamado de Iteração de Valor com Interpolação Linear (*CVaR Value Iteration with Linear Interpolation* – CVaR-VILI) para lidar com essas dificuldades. A primeira dificuldade é contornada com uso da interpolação linear, assim é feita uma discretização de Y. Além disso, foi explorada a concavidade de $yV(s,y)$ para delimitar o erro introduzido por essa técnica. A segunda dificuldade é contornada explorando a concavidade do problema de maximização para garantir que a otimização seja executada de forma eficaz.

Como mencionado, para se obter uma problema que seja tratável em tempo factível foi preciso discretizar Y definindo um conjunto de intervalos de confiança para calcular o operador de Bellman, i.e., foi definido um conjunto finito de valores de Y, e então interpola-se a função valor somente entre esses pontos de interpolação. Note que ao usar a interpolação linear para fazer esta aproximação é introduzido um erro de aproximação que pode ser maior ou menor dependendo do número de pontos de interpolação escolhido.

Seja $N(s)$ o número de pontos de interpolação e para todo s ∈ S, seja Y(s) = $\{y_1$, y_2, . . . , $y_{N(s)}$ $\} \in [0,1]^{N(s)}$ o conjunto de pontos de interpolação. Denotamos como $I_s[V](y)$ a interpolação linear de $yV(s,y)$ nesses pontos, que é definida por:

$$I_s[V](y) = y_i V(s,y_i) + \frac{y_{i+1}V(s,y_{i+1}) - y_i V(s,y_i)}{y_{i+1} - y_i}(y - y_i), \tag{11}$$

em que $y_i = max\{y' \in Y(s) : y' \leq y\}$ e $y_{i+1} = min\{y' \in Y(s) : y' \geq y\}$ de modo que y $\in [y_i, y_{i+1}]$; i.e, na interpolação são usados os dois pontos de interpolação mais próximos de y, chamados de y_i e y_{i+1}.

Uma vez que $I_{s'}[V]\big(y\xi(s')\big)$ é a interpolação linear de $yV\big(s', y \cdot \xi(s')\big)$, na equação 10 pode-se substituir $V\big(s', y\xi(s')\big)$ por $\frac{I_{s'}[V](y \cdot \xi(s'))}{y}$, obtendo o operador de Bellman Interpolado T_I, como mostrado a seguir Chow et al. (2015):

$$T_I[V](s,y) = \min_{a \in A}\{Q(s,y,a)\} = \min_{a \in A}\Big\{C(s,a) + \gamma \max_{\xi \in \mathcal{U}_{CVAR}(y,P(.|s,a))} \sum_{s' \in S} \frac{I_{s'}[V]\big(y \cdot \xi(s')\big)}{y} P(s'|s,a)\Big\}.$$

O conjunto Y de pontos de interpolação é escolhido pelo especialista de modo a balancear o desempenho e o menor erro de aproximação desejado. Note que, quanto maior o número de pontos de interpolação houver, menor será o erro de aproximação.

4. Algoritmo Exato de Avaliação de uma Política Estacionária

Embora o algoritmo CVaRVILI possa obter uma política ótima aproximada, ele possui um alto custo computacional devido a necessidade de resolver repetidas vezes problemas de programação linear. Por outro lado, o valor de uma política estacionária pode ser encontrado com custo computacional similar ao de um algoritmo de iteração de valor para MDP neutros ao risco. Nesta seção é proposto o algoritmo PECVaR (*Policy Evaluation* CVaR), que avalia uma política estacionária de forma exata. O algoritmo PECVaR baseia-se no seguinte teorema.

Teorema 2. *Considere uma política estacionária π e um MDP com custo constante, sem perda de generalidade com custo 1, tem-se que:*

$$CVaR_{\alpha=\left(1-P_G^T\right)}(Z|\pi) = \frac{\mathbb{E}[Z] - \mathbb{E}\left[Z|Z \leq \sum_{t=0}^{T-1}\gamma^t\right] P_G^T}{1 - P_G^T}, \tag{12}$$

em que $P_G^T = \Pr(s_T \in G|\pi)$ é a probabilidade de alcançar a meta em pelo menos T passos.

Demonstração. Note que, se a variável aleatória Z é contínua, a seguinte identidade é verdadeira Chow (2017):

$$CVaR_\alpha(Z) = \mathbb{E}[Z|Z > VaR_\alpha(Z)] \tag{13}$$

e utilizando a Lei da Probabilidade Total, tem-se que:

$$\begin{aligned}\mathbb{E}[Z] = \;& \mathbb{E}[Z|Z > VaR_\alpha(Z)]\Pr(Z > VaR_\alpha(Z)) \\ & +\mathbb{E}[Z|Z \leq VaR_\alpha(Z)]\Pr(Z \leq VaR_\alpha(Z))\end{aligned} \tag{14}$$

e implica em:

$$\mathbb{E}[Z|Z > VaR_\alpha(Z)] = \frac{\mathbb{E}[Z] - \mathbb{E}[Z|Z \leq VaR_\alpha(Z)]\Pr(Z \leq VaR_\alpha(Z))}{\Pr(Z > VaR_\alpha(Z))}. \tag{15}$$

No caso em que o custo é unitário, tem-se que:

$$\Pr\left(Z \leq \sum_{t=0}^{T-1}\gamma^t\right) = P_G^T \tag{16}$$

e

$$VaR_{\alpha=\left(1-P_G^T\right)}(Z) = \sum_{t=0}^{T-1}\gamma^t. \tag{17}$$

Juntando as equações 13, 15, 16 e 17, obtém-se o resultado do teorema. □

O algoritmo PECVaR (Algoritmo 1) utiliza a equação 12 para calcular o valor de todos os estados $s \in S$, considerando cada um deles como estado inicial, e para alguns valores de α, as probabilidades no formato $1 - P_G^T$ (linha 18).

O algoritmo PECVaR itera em valores de $T = \{0, 1, 2, \ldots\}$ e atualiza em cada iteração os valores para $\Pr(s_T = s' | s_0 = s, \pi)$ (linha 21), $P_G^T(s)$ (linha 23) e $\mathbb{E}\left[Z | Z \leq \sum_{t=0}^{T-1} \gamma^t, s_0 = s, \pi\right]$ (linha 25). As atualizações são obtidas por:

$$\Pr(s_T = s' | s_0 = s, \pi) = \sum_{i \in S} P(s' | i, \pi(s)) \Pr(s_{T-1} = i | s_0 = s, \pi) \tag{18}$$

$$P_G^T(s) = \sum_{s' \in G} \Pr(s_T = s' | s_0 = s, \pi) \tag{19}$$

e

$$\mathbb{E}\left[Z | Z \leq \sum_{t=0}^{T-1} \gamma^t, s_0 = s\right] = \frac{\mathbb{E}\left[Z | Z \leq \sum_{t=0}^{T-2} \gamma^t, s_0 = s\right] P_G^{T-1}(s) + \sum_{t=0}^{T-1} \gamma^t (P_G^T(s) - P_G^{T-1}(s))}{P_G^T(s)}. \tag{20}$$

Finalmente, o algoritmo PECVaR percorre apenas alguns valores de α, mas valores que não foram percorridos podem também ser calculadas de forma exata, basta ponderar o valor CVaR calculado com o melhor custo acumulado considerado (linha 28 e 29).

5. Experimentos

Foram realizados experimentos comparando o algoritmo CVaRVILI com diferentes inicializações em dois domínios, o domínio *Fast-Slow* e o domínio *Grid World* descritos a seguir.

Domínio Fast-Slow. No domínio *Fast-Slow* existe um caminho reto representado por um grid $N \times 1$. O agente parte do início do caminho e seu objetivo é chegar ao final do caminho. Em cada estado o agente pode escolher uma de duas ações possíveis: (i) a ação *fast* que faz com que o agente se movimente para o próximo estado com 75% de probabilidade e para o estado anterior com 25% de probabilidade; e (ii) a ação *slow* que permite que o agente se movimente para o próximo estado com 50% de chance ou permaneça no mesmo estado com 50% de probabilidade. O custo de qualquer ação é 1.

Domínio Grid World. O domínio *Grid World* Chow et al. (2015) tem um mapa de terreno 2D que é representado por um grid $N \times M$. Um agente (por exemplo, um veículo robótico) começa em uma região segura e seu objetivo é viajar para um determinado destino. Em cada passo, o agente pode se mover para qualquer uma das suas posições vizinhas. Porém, existe uma probabilidade p de se movimentar para um estado vizinho aleatório. O custo para se mover de um estado para o outro e que está associado ao uso de combustível é 1. Entre o ponto de partida e o destino, há uma série de obstáculos que o agente deve evitar pois se o agente bater nele, ele ficará quebrado sem a possibilidade de

Algoritmo 1 Policy Evaluation valor CVaR (PECVaR)

Entrada: Um MDP(S,A,P,C,γ), Y (s) = $\{y_1, y_2, \ldots, y_{N(x)}\} \in [0,1]^{N(s)}$**,** $maxIter$**,** $minAlpha$**,** $targetS$**, e** π
Saída: função CVaR V_{CVaR}
i $\leftarrow 1$;
$V_\pi(s) \leftarrow 0, CVaR(s,0) \leftarrow 0, \forall s \in S$;
enquanto $i < maxIter$ **faça**
    para cada $s \in S$ **faça**
        $V(s) \leftarrow C(s,\pi(s)) + \gamma \sum_{s'} V(s')P(s'|s,\pi(s))$
        $CVaR(s,0) \leftarrow C(s,\pi(s)) + \gamma \max_{s' \in \{x|P(x|s,\pi(s))>0\}} CVaR(s',0)$
    fim
    $i \leftarrow i+1$
fim
para cada $s \in S$ **faça**
    $T \leftarrow 0, P_G^0 \leftarrow 0, C^0 \leftarrow 0, V_{\leq C}^0 \leftarrow 0, P_s^0 \leftarrow 1, P_{s'}^0 \leftarrow 0 \; \forall s' \neq s \in S$
    enquanto $1 - P_G < minAlpha$ **faça**
        $CVaR(s, 1-P_G^T) \leftarrow \dfrac{V(s) - V_{\leq C}^T \times P_G^T}{1 - P_G^T}$
        $T \leftarrow T+1$
        para cada $s' \in S$ **faça** $P_{s'}^T \leftarrow \sum_{i \in S} P(s'|i,\pi(s))P_i^{T-1}$
        $P_G^T \leftarrow \sum_{s' \in G} P_{s'}^T$
        $C^T \leftarrow C^{T-1} + \gamma^{T-1}$
        $V_{\leq C}^T \leftarrow \dfrac{V_{\leq C}^{T-1} \times P_G^{T-1} + C \times \left(P_G^T - P_G^{T-1}\right)}{P_G^T}$
    fim
    para cada y $\in$ Y **faça**
        $T \leftarrow \min\{t | y \geq 1 - P_G^t\}$
        $V_{CVaR}(s,y) \leftarrow \dfrac{(1-P_G^T) \times CVaR(s, 1-P_G^T) + (y - (1-P_G^T)) \times C^T}{y}$
    fim
fim
devolva V_{CVaR};

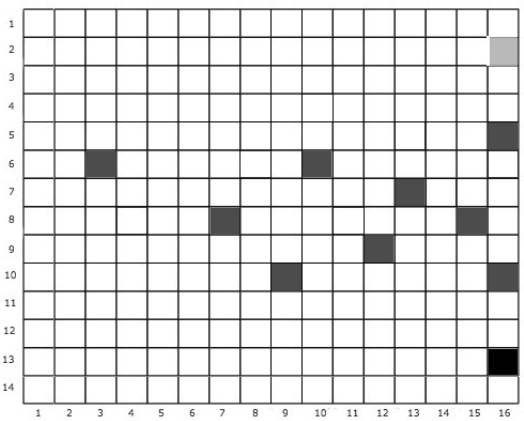

Figura 1. Instância do *GridWorld* de tamanho 14×16. O estado objetivo é o cinza claro, o estado inicial é preto e os obstáculos são cinza escuro.

chegar no estado objetivo. A Figura 1 mostra uma instância do domínio *GridWorld* com uma grid de tamanho 14×16.

O objetivo dos experimentos é investigar a possibilidade de melhoria do algoritmo CVaRVILI utilizando o algoritmo PECVaR durante a inicialização do mesmo. Três abordagens diferentes foram utilizadas para inicialização do algoritmo CVaRVILI, a primeira utiliza a função valor $V_0 = 0$ (chamada V_0), a segunda utiliza a função valor obtida pelo algoritmo de *Iteração de Valor* para MDPs neutros a risco (chamada V_{mean}) e a terceira utiliza a política obtida pelo algoritmo de *Iteração de Valor* para MDPs neutros a risco e o algoritmo PECVaR para calcular a função valor dessa politica (chamada V_{pecvar}).

Foi calculado o tempo total de execução do CVaRVILI considerando as diferentes abordagens de inicialização. No caso da abordagem V_0, o tempo de execução inclui apenas o custo computacional do algoritmo CVaRVILI. No caso da abordagem V_{mean} soma-se (i) o tempo gasto do algoritmo de *Iteração de Valor*; e (ii) o tempo de execução até convergência do algoritmo CVaRVILI. No caso da abordagem do V_{pecvar} soma-se: (i) o tempo gasto do algoritmo de *Iteração de Valor* para encontrar a política ótima do MDP neutro a risco; (ii) o tempo de execução do algoritmo PECVaR para calcular o valor da política encontrada pelo algoritmo de *Iteração de Valor*; e (iii) o tempo de execução até convergência do algoritmo CVaRVILI. Para os experimentos foram utilizadas três quantidades diferentes de pontos de interpolação: 11, 21 e 31. O fator de desconto utilizado em todos os casos foi $\gamma = 0.95$ e $\epsilon = 1e - 3$.

Na figura 2 são mostrados os resultados para duas instâncias do domínio *Fast-Slow*, uma de 7 estados e outra de 70 estados. Para a instância *Fast-Slow* de 7 estados o tempo gasto do algoritmo de *Iteração de Valor* foi 0.015 segundos e o tempo de execução do algoritmo PECVaR para as diferentes quantidades de pontos de interpolação foi menor ou igual que 0.034 segundos. Para a instância *Fast-Slow* de 70 estados o tempo gasto do algoritmo de *Iteração de Valor* foi 0.21 segundos e o tempo de execução do algoritmo PECVaR para as diferentes quantidades de pontos de interpolação foi menor ou igual que 0,24 segundos. Neste domínio, a abordagem de inicialização com V_{pecvar} teve desempenho melhor em todos os casos, sendo até 5.7 vezes mais rápida que a inicialização com V_0 (no experimento com 7 estados e com 31 pontos de interpolação). Enquanto que a abordagem de inicialização V_{mean} foi pior em 4 dos 6 experimentos do que a abordagem de inicialização V_0.

Na figura 3 são mostrados os resultados para duas instâncias do domínio Grid World, uma de 224 estados e outra de 3392 estados. Para a instância com 224 estados, o tempo gasto do algoritmo de *Iteração de Valor* foi 1.72 segundos e o tempo de execução do algoritmo PECVaR para as diferentes quantidades de pontos de interpolação foi menor ou igual que 0.48 segundos. Para a instância com 3392 estados, o tempo gasto do algoritmo de *Iteração de Valor* foi 165.63 segundos e o tempo de execução do algoritmo PECVaR para as diferentes quantidades de pontos de interpolação foi menor ou igual que 336.21 segundos. Neste domínio, a abordagem de inicialização com V_{mean} foi a melhor em 5 dos 6 casos, enquanto que a abordagem de inicialização V_{pecvar} foi a melhor em 1 dos 6 casos.

Note que, o tempo gasto na execução dos algoritmos *Iteração de Valor* e PECVaR é muito menor do que o tempo de execução do algoritmo CVaRVILI. Nos domínios tes-

tados o tempo gasto de execução do algoritmo PECVaR foi apenas até 2 vezes maior do que o algoritmo de *Iteração de Valor* para MDP neutros ao risco.

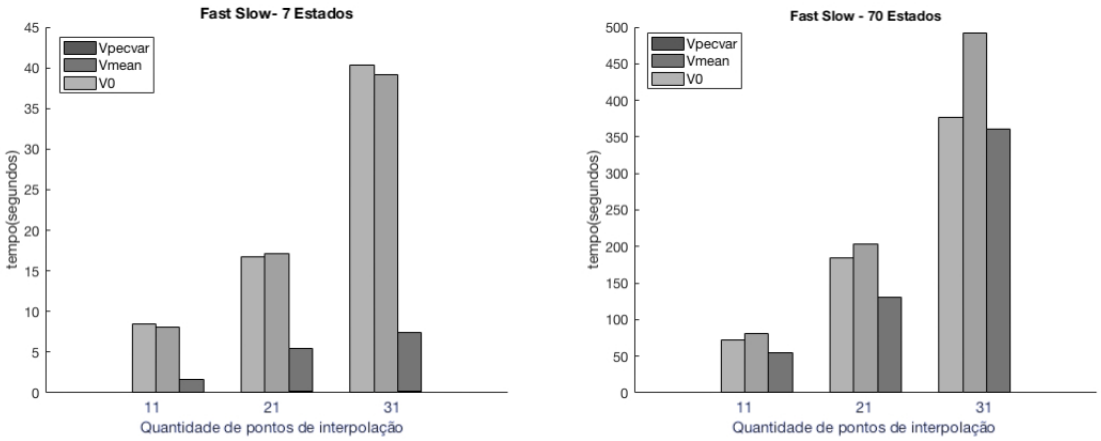

Figura 2. Tempo total de execução do algoritmo CVaRVILI com diferentes abordagens de inicialização para duas instâncias do domínio *Fast-Slow* com 11, 21 e 31 pontos de interpolação.

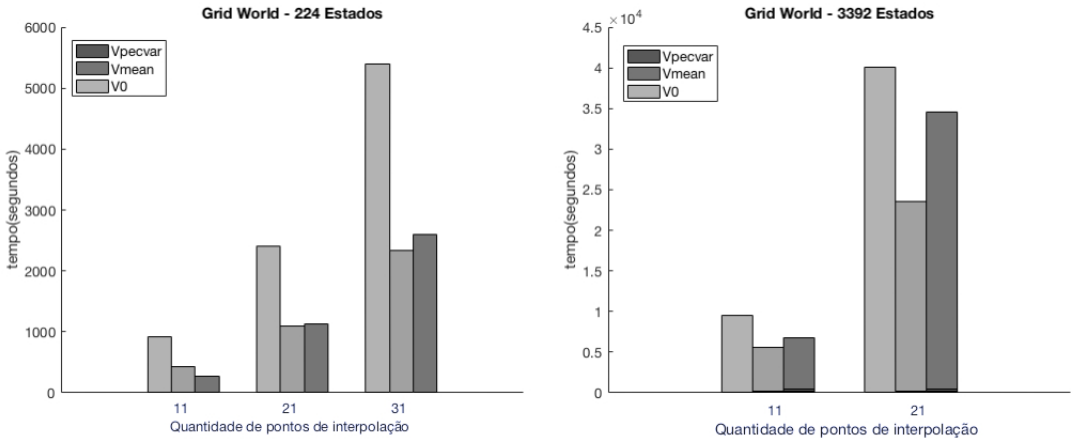

Figura 3. Tempo total de execução do algoritmo CVaRVILI com diferentes abordagens de inicialização para a instância com 224 estados do domínio *Grid World* com 11, 21 e 31 pontos de interpolação e para a instância com 3392 estados com 11 e 21 pontos de interpolação.

6. Conclusão

Existe um aumento do uso da métrica CVaR no contexto de tomada de decisão sequencial como uma forma de garantir robustez, em relação a estocasticidade e os erros de modelagem. Porém, apesar dos avanços e dos resultados obtidos por alguns trabalhos que utilizam a função CVaR em MDPs, existe uma carência de trabalhos que mostrem resultados experimentais abrangentes avaliando esse novo critério.

Neste trabalho é proposto o Algoritmo PECVaR que é um método eficiente para calcular o valor de uma política estacionária de custo constante de MDPs que utilizam a métrica CVaR. Diferente do algoritmo CVaRVILI, o algoritmo PECVaR não precisa resolver nenhum problema linear e portanto naõ é tão custoso computacionalmente como

o algoritmo CVaRVILI. Além disso, foram realizados experimentos comparando abordagens diferentes para inicialização do Algoritmo CVaRVILI. Os experimentos demonstraram que a abordagem com inicialização com V_{pecvar} foi até 5.7 vezes melhor no tempo total de execução comparado com a abordagem que utiliza a inicialização V_0.

Para os trabalhos futuros almeja-se generalizar o PECVaR para computar políticas estacionárias de custo não constante e testar o algoritmo em outros domínios. Por outro lado, os resultados com as diferentes abordagens de inicialização corroboram a ideia de que o algoritmo CVaRVILI tem como ser melhorado, pode ser com uma boa abordagem de inicialização ou com outras técnicas computacionais como por exemplo paralelização que pode trazer significativas melhorias no desempenho do CVarVILI, o que pode abrir caminhos para o processamento de problemas ainda maiores em tempos factíveis. Assim, um outro possível trabalho futuro é o uso de paralelização no CVaRVILI.

Agradecimentos

Os autores agradecem à FAPESP pelo apoio financeiro (processo #2018/11236-9).

Referências

Bellman, R. E. (1957). Dynamic programming. *Princeton University Press*.

Carpin, S., Chow, Y.-L., & Pavone, M. (2016). Risk aversion in finite Markov decision processes using total cost criteria and average value at risk. In *2016 ieee international conference on robotics and automation (icra)* (pp. 335–342).

Chow, Y. (2017). *Risk-sensitive and data-driven sequential decision making* (Unpublished doctoral dissertation). Stanford University.

Chow, Y., Tamar, A., Mannor, S., & Pavone, M. (2015). Risk-sensitive and robust decision-making: a CVaR optimization approach. *Advances in Neural Information Systems*, 1522–1530.

Iyengar, G., & Ma, A. K. C. (2013). Fast gradient descent method for mean-CVaR optimization. *Annals of Operations Research*, *205*(1), 203–212.

Pflug, G. C., & Pichler, A. (2016). Time-consistent decisions and temporal decomposition of coherent risk functionals. *Mathematics of Operations Research*, *41*(2), 682–699.

Puterman., M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York, NY: Wiley-Interscience.

Uryasev, S., & Rockafellar, R. T. (2001). Conditional value-at-risk: Optimization approach. *Stochastic Optimization: Algorithms and Applications*, 411–435.