Aprendizado de máquina por reforço aplicado no Jogo de Cartas Uno

Pedro Harmendani
Departamento de Informática
Instituto Federal de Educação Ciência e Tecnologia do Sudeste de Minas Gerais
Rua Bernardo Mascarenhas, 1283
Juiz de Fora Minas Gerais Brasil
pedrohnh@hotmail.com

Márcia Zanetti
Departamento de Informática
Instituto Federal de Educação Ciência e Tecnologia do Sudeste de Minas Gerais
Rua Bernardo Mascarenhas, 1283
Juiz de Fora Minas Gerais Brasil
marcia.zanetti@ifsudestemg.edu.br

ABSTRACT
Stochastic and complex environments such as card games represent a real challenge for Artificial Intelligence. This article proposes the application of the Q-learning algorithm, a popular machine learning technique by reinforcement used in deterministic and non-deterministic scenarios. In order to validate this application in the proposed scenario, it was necessary to develop a computational platform faithful to the game environment in which artificial agent models were deployed in order to build an agent with superior performance. Experiments have shown that the artificial player developed has achieved fifty-three percent average wins over several match iterations simulated on the proposed platform.

Keywords
Reinforcement Learning, Q-Learning; Artificial Intelligence, Stochastics Process; Cards Game.

1. INTRODUÇÃO
Em Inteligência Artificial (IA), o Aprendizado por Reforço (AR) é um campo do Aprendizado de Máquina (AM) que se baseia na aprendizagem por feedback através da interação do agente artificial com o seu ambiente. Um processo de AR permite modelar um conjunto de ações e estados de tal forma que as recompensas ou punições recebidas pelo agente ao longo do tempo possam ser associadas a tomada de decisão. (Littman, 2015).

O AR possui diversas aplicações retratadas em pesquisas, como em jogos de estratégia em tempo real (Sethy, 2015; Neto, 2013), no problema da mochila multidimensional (Ottoni, 2017), na navegação de robôs sob ambiguidade sensorial em ambientes dinâmicos de natureza estocástica (Monteiro, 2004) e no jogo de Gamão em que bots foram capazes de vencer os melhores jogadores de Gamão do mundo (Tesauro, 1995).

No contexto de jogos de cartas, a definição das melhores estratégias de jogada, bem como a grande quantidade de estados e a sua natureza estocástica, evidencia-se um problema desafiador para a Inteligência Artificial, pois a busca por soluções eficientes nesse tipo de ambiente ainda se trata de uma pesquisa com amplas possibilidades. (Mendes, 2008; Ferreira, 2008; Fazio, 2008; Pereira, 2012; Sethy, 2015; Bravi, 2019).

O AR é um conceito e ao mesmo tempo um típico problema de aprendizagem que permite a modelagem de agentes artificiais que retém conhecimento a partir de situações vivenciadas em ambientes desconhecidos e não determinísticos, semelhantes aos jogos de cartas, o que torna essa abordagem apropriada para ambientes em que se tem elevado número de estados e um fator de estocasticidade nas transições entre estados (Sutton, R. S. e Barto, A. G, 2017).

Genericamente, problemas de AR podem ser resolvidos por duas estratégias, a primeira, conhecida por programação genética e a segunda, por métodos estatísticos combinados com técnicas de programação dinâmica (Kaelbling, 1996). A segunda abordagem foi adotada neste trabalho, visto que ainda não há na literatura a definição da melhor estratégia aplicada nessas circunstâncias. Sendo assim, cabe avaliar a viabilidade de implementação de técnicas do AR nesse cenário com construção de agentes artificiais jogadores que aprendam a tomar as melhores decisões durante as partidas de jogos de cartas, tornando-as emocionantes e mais desafiadoras.

Neste artigo foi proposta a utilização do algoritmo Q-Learning (Watkins, 1989), aplicado ao popular jogo de cartas Uno. Essa metodologia é uma tradicional técnica algorítmica de AR que maximiza o valor da recompensa futura recebida pelo agente ao longo do tempo, criando uma política ótima de ações sob certas condições (Sutton, R. S. e Barto, A. G., 2017).

No cenário onde um agente artificial atua no ambiente de um jogo de cartas Uno, o objetivo é ampliar seu desempenho diante de outro adversário artificial, munido de inteligência básica implementada por meio de um sistema especialista, para tomada de decisões utilizando critérios gulosos para seleções de ações. Nesse experimento foi possível evidenciar o grau de relevância do algoritmo quando aplicado às características estocásticas do ambiente ao avaliar o aprendizado do agente artificial ao longo de diversas simulações.

2. APRENDIZADO POR REFORÇO
Antes de descrever o aprendizado de máquina estudado neste artigo é necessário compreender que existem basicamente alguns tipos de aprendizado de máquina na literatura: aprendizado supervisionado, não supervisionado e através de reforço. O primeiro envolve a supervisão das ações, baseado no quanto o resultado obtido se aproxima do esperado. A partir da manipulação realizada nos dados de entrada, o agente aprende a regra geral com base na aceitação, indicado por uma supervisão externa. No aprendizado de máquina não supervisionado não é

utilizado uma fonte externa, como um supervisor no processo de aprendizado. Nesse caso, o agente não precisa de um modelo de dados de saídas e entradas pré-definidos. Isso também ocorre no aprendizado por reforço já que o aprendizado é estabelecido ao explorar algum ambiente desconhecido, porém, o processo se concentra em maximizar alguma noção de recompensa ao receber um feedback do ambiente após a execução de uma determinada ação (Russel e Norvig, 2013).

A base teórica para o caso geral do AR se fundamenta na definição formal do processo decisório Markoviano visto que as ações tomadas pelo agente não apenas determinam a recompensa recebida, mas também o próximo estado do ambiente. Em outras palavras, um ambiente que satisfaz um modelo de processo de decisão Markov (MDP - Markov Decision Process) tem a capacidade de prever quais os estados e recompensas futuras do ambiente a partir do estado presente sem depender dos estados anteriores, essa é a propriedade fundamental que permite um domínio de problemas serem formulados sob as condições do MDP (Watkins, 1989 ; Kaelbling, 1996).

A fim de explicitar uma notação clara e formal que permeará as elucidações futuras neste artigo é apresentada a nomenclatura base a partir do MDP que pode ser definido formalmente como uma tupla <S, A, P, R> em que S representa o conjunto de estados finito do ambiente e um conjunto de ações A. O conjunto de estados e ações definem também: P que é a probabilidade (s′|s, a) tal que st ∈ S e at ∈ A. E o reforço r ∈ R associa a cada ação a ∈ A tomada em um tempo discreto no estado s ∈ S (Littman, 2015).

Segundo Kaelbling (1996), os seguintes passos definem o processo de aprendizado por reforço:

1. O agente observa o ambiente (Computa st ∈ S).

2. Uma política arbitrária seleciona uma determinada ação at ∈ A.

3. O agente executa a ação at em um tempo discreto t qualquer.

4. O agente recebe um sinal de reforço r ∈ R.

5. A informação de recompensa associada ao par estado/ação correspondente é atualizada.

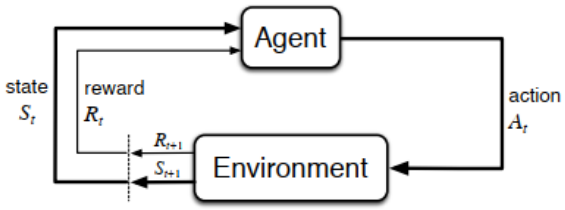

Figura 1. Descreve o modelo de AR em um ambiente MDP.

A figura 1 descreve o modelo de AR em um ambiente MDP (Sutton, 2017). Esse modelo algorítmico caracteriza o problema central do aprendizado por reforço que consiste em escolher a melhor ação para o agente em um estado arbitrário definido em S. Isso significa que o agente deve aprender a escolher as ações que garantirão o maior sinal de reforço recebido, e desde então, seguir uma política que assegurará o melhor feedback futuramente. Essa escolha sequencial de ações visando obter o melhor sinal de reforço é definida como sendo uma política ótima de ações para o agente, encontrá-la pode ser algo não trivial ou até computacionalmente impossível, sobretudo em ambientes estocásticos e parcialmente observáveis (Watkins,1989).Em ambientes estocásticos e onde não exista um modelo que o descreva suficientemente, isso é, onde não há um modelo capaz de definir aspectos importantes do ambiente para a construção de uma política consistente de ações, utiliza-se algoritmos denominados na literatura de model-free. Essas técnicas se consagram como as mais populares e fáceis de implementar em ambientes complexos, visto que uma função de transições entre estados não precisa ser previamente conhecida. Portanto, o sucesso do aprendizado nessas condições, depende do (treinamento) do agente no ambiente ao explorar o espaço de estados e ações uma quantidade suficiente de vezes (Kaelbling, 1989).

A aplicação dos conceitos formais de AR aplicadas em cenários não determinísticos e parcialmente observáveis, como no jogo de cartas Poker (Fazio, 2008) e em jogos de estratégia em tempo real, comprovam a eficácia do AR através de experimentos empíricos na utilização de técnicas algorítmicas como o SARSA (Sethy, 2015) e o Q-Learning (Monteiro, 2004). Em alguns casos, os resultados obtidos por pesquisas empíricas podem não encontrar uma política ótima de ações ou ser estabelecido formalmente como a melhor técnica para um determinado problema, todavia, demonstram-se ser benéficos quando aplicados em uma variedade de problemas (Ottoni, 2017; Sethy, 2015; Neto, 2013 ; Benicasa, 2012; Selvatici, 2005; Faria e Romero, 1999).

2.1 O Q-Learning

O Q-Learning (Watkins,1989) é um popular algoritmo aplicado para o aprendizado automático de agentes classificado na literatura como um algoritmo livre de modelo - isso significa na prática que os estados transitados pelo agente não são dados por uma função ou distribuição de probabilidade conhecida, mas através de múltiplas interações com o ambiente e que, sob certas condições, será capaz de determinar os estados futuros, e portanto, aplicar transições que garantirão as melhores recompensas. Trata-se de um algoritmo que permite iterativamente aprender uma política ótima de ações em ambientes modelados como processos decisórios Markovianos determinísticos ou não-determinísticos, de forma que a convergência do aprendizado possa ocorrer em ambos os cenários (WATKINS, 1989).

O algoritmo Q-Learning e descrição das variáveis são abordados a seguir (Kaelbling, 1996):

$$Q(s,a)= Q(s,a)+ \alpha(r+ \gamma Q(s',a ')-Q(s,a)) \quad (1).$$

Para melhor elucidar a regra de atualização do algoritmo 1, temos que:

Taxa de aprendizagem α: determina qual fração da antiga estimação do valor Q(s,a) será atualizada com o novo valor Q. O fator α é maior que 0 e geralmente menor que 1. Fator de desconto γ (onde $0 \leq \gamma \leq 1$): determina a importância das recompensas futuras. Função de Valor dada por Q(s,a) indica o valor Q dado o estado atual s e a ação a executada em s. Sendo que s' representa o estado futuro. Portanto, o valor maxQ(s',a ') representa o maior valor de ação na função Q para o próximo estado dado s'.

Em ambientes não determinísticos, para garantir a convergência do algoritmo, a regra de atualização de Q é dada por:

$$Q(s,a)=(1-\alpha)Q(s,a)+ \alpha(r+ \gamma Q(s',a ')-Q(s,a)) \quad (2).$$

No caso mencionado, o fator de aprendizagem α é definido por:

$$a=1/(n(S)) \quad (3),$$

tal que n representa a quantidade de vezes que s foi visitado pelo agente.

Uma vantagem interessante do método, que o torna popular, é que a convergência do aprendizado não depende das políticas de exploração adotadas para o agente. Todavia, devem ser abordadas para que o agente possa explorar o ambiente e adquirir novas experiências ao se deslocar para estados poucos visitados ou desconhecidos (Kaelbling, 1996).

Pesquisas realizadas em ambientes totalmente desconhecidos demostram a capacidade do método. Experiências empíricas confirmam a sua viabilidade de utilização na navegação autônoma de robôs, visto que o tempo de convergência do aprendizado para uma política ótima de ações neste cenário não-determinístico obteve resultados favoráveis (Benicasa, 2012). Monteiro (2004) também propõe um experimento conduzido sobre ruídos sensoriais na navegação de robôs atuando em um ambiente parcialmente observável, violando, portanto, a condição de Markov, e impossibilitando o agente de encontrar uma política ótima de ações. Os resultados da pesquisa demonstram que o Q-Learning foi o algoritmo que obteve os melhores resultados ao encontrar uma política subótima em um intervalo de tempo menor que outros métodos como o SARSA e o Q(λ) (Peng, 1996).

3. O MODELO PROPOSTO

O trabalho proposto visa aplicar o algoritmo Q- Learning no aprendizado de um agente artificial no jogo de cartas Uno de modo a aprimorar suas ações e tornar suas jogadas cada vez mais imprevisíveis perante um outro jogador artificial sem AR; - dotado de inteligência básica na tomada de decisões utilizando um sistema especialista para escolha das suas ações. Para isso, foi desenvolvido um ambiente computacional específico capaz de realizar simulações das partidas do jogo Uno e que também permite obter dados oportunos para a conclusão deste trabalho.

O ambiente de simulação citado foi modelado a partir de alguns princípios fundamentais que norteiam o objetivo da proposta, são eles: um conjunto discreto {S} de estados do ambiente, um conjunto discreto de ações {A} do agente e um sinal de reforço definido por um escalar ou real gerado por uma função. Formalmente, esses princípios constituem a base de conjuntos discretos necessários para a construção de um modelo de aprendizado por reforço modelados sob os princípios do MDP (Kaelbling, 1996).

Para detalhar o ambiente computacional desenvolvido e validar os resultados obtidos com as experimentações, o modelo do ambiente de jogo e os detalhes da técnica algorítmica de AR utilizada para o aprendizado do agente são elucidadas adiante.

3.1 O jogo de cartas Uno

O objetivo do jogo de Cartas Uno consiste em ficar sem cartas na mão, para tal, o jogador poderá utilizar todos os recursos disponíveis no jogo para impedir que os adversários façam o mesmo. O jogador consegue realizar o descarte ao combinar as cartas em mãos com a carta da mesa através da mesma cor, número ou símbolo. As cartas do tipo Curinga podem ser descartadas sobre qualquer carta.

O jogo de cartas Uno sempre começa com todo o montante de 108 cartas embaralhadas. A partir disso é distribuído sete cartas para cada jogador, ficando o restante das cartas com as faces voltadas para baixo. A última carta desse montante é colocada na "mesa" tendo a face voltada para cima. A partir disso, um jogador fará a sua primeira jogada dando início a uma sequência de jogadas de cartas na mesa por todos os jogadores até que algum jogador consiga ficar sem cartas nas mãos – considerado o vencedor da partida.

A plataforma de simulações desenvolvida para esta pesquisa se baseia nas regras clássicas e na distribuição de cartas oficial do jogo Uno. No desenvolvimento da plataforma computacional do jogo foi necessário abstrair e categorizar as cartas do jogo bem como definir a contagem correta das cartas. A seguir, detalhamos o tipo e a contagem das cartas:

Cartas do tipo Normal:

19 cartas azuis - de 0 a 9 ** somente 1 carta é zero.

19 cartas verdes - de 0 a 9.

19 cartas vermelhas - de 0 a 9.

19 cartas amarelas - de 0 a 9.

Cartas do tipo Ação:

8 cartas "Compra duas cartas" - duas de cada cor.

8 cartas "Salta" - duas de cada cor.

8 cartas "Inverte" - duas de cada cor.

Cartas do tipo Curinga.

4 cartas "Curinga normal".

4 cartas "Curinga compra 4 cartas".

Para valorizar as cartas do baralho foi definido o peso padrão clássico adotado no jogo para cada tipo de carta. As cartas mais valorizadas são as cartas Curinga que valem 50 pontos, cartas de Ação valem 20 pontos. Por último, as cartas normais que possuem o valor indicado pelo número da sua face.

3.2 Os agentes e o mecanismo de exploração

Com o intuito de validar e alcançar o objetivo desta pesquisa, foram implementados agentes inteligentes capazes de atuar no ambiente proposto. Dois agentes distintos foram propostos, o primeiro agente utilizou a metodologia baseada no aprendizado de máquina por reforço e o outro, foi munido de um sistema especialista para selecionar as melhores ações por meio de uma estratégia gulosa como mecanismo de atuação no ambiente. Ambos possuem as mesmas limitações de escopo quanto as regras do jogo Uno, bem como o mesmo espaço de estados e ações.

O dilema da exploração do ambiente pelo agente é um tema recorrente de pesquisas em IA, mais propriamente em aprendizado de máquina por reforço. Existem diversas técnicas e métodos de exploração na literatura, no entanto, alguns métodos são mais adequados para problemas específicos, sendo alguns, comprovados formalmente. Outros métodos de exploração não são comprovados a rigor matemático, classificados como técnicas Ad-hoc de exploração. Entretanto, são populares por apresentar resultados razoáveis na prática e computacionalmente viáveis.

Tendo em vista a implementação de uma plataforma de simulação de partidas do jogo de cartas Uno, a escolha por utilizar uma heurística que se baseia na exploração aleatória do ambiente - uma técnica Ad-hoc, justifica-se por ser facilmente implementada e testada em um ambiente com o alto fator de entropia. Além disso, o método de aprendizado Q-Learning, utilizado nesta pesquisa, não depende do mecanismo exploratório para otimizar uma política de ações (Kaelbling, 1996). Adiante, explicitamos o mecanismo exploratório do agente e suas peculiaridades.

O agente com AR possui um mecanismo para explorar o ambiente e não cair em mínimos ou máximos locais da função de

aprendizado. Sendo assim, o fator de exploração é definido por um número natural arbitrário que representará os intervalos em que o agente executará uma ação não baseada no conhecimento através do aprendizado. Para fins práticos, considera-se uma taxa de exploração qualquer igual a dez, significa que a cada dez rodadas do jogo, uma ação seguinte do agente será executada por um mecanismo exploratório descrito mais adiante. Formalmente, a taxa de exploração ε é estabelecida em função do intervalo de rodadas In de uma partida, sendo n o representativo ordinal da enésima rodada do jogo:

$$\varepsilon=I\_n+1. \quad (4).$$

Após a execução de uma ação usando esse mecanismo, temos que I_n valerá zero. Por conseguinte, a cada outra nova rodada do jogo em sequência, temos que: I_n=I_n+1.

Esse mecanismo pode selecionar uma ação aleatória ou utilizar o sistema especialista do agente sem aprendizado para executar uma ação. Temos que a probabilidade de selecionar uma dessas políticas foi definida em 50% para todo intervalo definido pelo fator de exploração.

3.3 Os estados e ações do ambiente

Como o jogo possui 108 cartas, foi preciso abstrair algumas dessas diferentes perspectivas do ambiente de jogo sobre as quais será possível criar um espaço de estados finito e computacionalmente viável em função dos artefatos de computação utilizados no desenvolvimento e simulações do modelo construído.

Para melhor representar essa abstração de estados e em uma tentativa de aproximar o modelo desenvolvido com um ambiente observável, foi definido um conjunto de três subestados denominado de Visão. Essa abordagem criada visa garantir que diferentes perspectivas visíveis do jogo possam contribuir para uma representação de estado mais fiel com a realidade, e, portanto, atribuindo consistência ao aprendizado do agente em situações reais semelhantes.

Uma descrição em alto nível das visões do ambiente e valores formais representativos podem ser vistos a seguir:

Visão Mesa: verifica qual é o tipo de carta atualmente na mesa de jogo:

x1 – A carta e do tipo Normal.

x2 – A carta é do tipo Ação.

x3 – A carta é do tipo Curinga.

Visão Contagem: realiza contagem das cartas da pilha de descarte e do montante de compra de cartas.

y1 – Quando a quantidade de cartas Curinga é igual a quatro ou doze cartas do tipo ação foram descartadas na pilha de descarte.

Senão:

y2 – Se a quantidade de cartas Curinga é maior que zero ou há pelo menos seis cartas do tipo Ação já descartadas.

y3 – Nenhuma das condições anteriores ocorreram.

Visão das Mãos: Realiza a contagem de cartas das mãos do jogador atual e do seu adversário.

z1 – A quantidade de cartas do adversário é menor.

z2 – A quantidade de cartas dos jogadores é igual.

z3 – A quantidade de cartas em mãos é inferior a quantidade do adversário.

A princípio, admite-se $\forall$ x,y,z $\in${A},{B} e {C}, respectivamente, tal que:

$$A \cap B \cap C= \emptyset \quad (5).$$

Um estado S qualquer é composto pela seguinte tupla: (xi, yi, zi) $\forall i \in N$.

Sendo assim, o espaço de estados possíveis do ambiente é dado pela combinação de seus subestados:

$$3\text{^}n= n(A) \times n(B) \times n(C) \quad (6).$$

Portanto, o espaço de estados possíveis do ambiente se limita em 27 possibilidades.

Devido ao custo temporal e computacional envolvidos na implantação e desenvolvimento do ambiente utilizado nesta pesquisa empírica, tornou-se relevante abstrair do total de jogadas possíveis e de eventuais estratégias, algumas ações mais simples, no entanto, mais objetivas ao se basearem na contagem e peso das cartas já definidos anteriormente neste artigo.

É descrito a seguir, um conjunto de quatro ações disponíveis para os agentes jogadores no ambiente proposto, elas são divididas em duas estratégias primárias:

Estratégia de seleção por valor da carta: essa primeira estratégia, a mais simples, consiste em utilizar o valor das cartas para tomar decisões. Subdividem-se em duas ações para o agente:

Ofensiva: O jogador escolhe as cartas de maior peso para serem jogadas na mesa. A seguinte ordem de prioridade de descarte por tipo de carta é definida para esta ação:

1. Curinga
2. Ação
3. Normal

Defensiva: O jogador escolhe as cartas de menor peso para serem jogadas na mesa. A seguinte ordem de prioridade de descarte por tipo de carta é definida para esta ação:

1. Normal
2. Ação
3. Curinga

Estratégia de seleção por índice de descarte: essa abordagem utiliza fatos estatísticos coletados no ambiente para ser efetuada e se subdivide em duas ações concretas:

Descarte por cor: a carta é escolhida pela cor que possui o maior índice de descarte dada a sua cor considerando as cartas da pilha de descarte e das mãos do jogador.

Descarte por número: a carta é escolhida pelo número que possui maior índice de descarte considerando o montante de descarte e a lista de cartas em mãos do jogador.

É importante salientar que no decorrer das rodadas do jogo, nem todas as ações estarão disponíveis já que as ações dependem fortemente do tipo de carta presente nas mãos do jogador e da carta da mesa. No modelo proposto, podem ocorrer cenários perfeitos em que o jogador só possuirá uma jogada disponível a ser efetuada.

3.4 A função de recompensa

A recompensa e a punição são aspectos biológicos dos seres, sendo abstraídos para o AR como um sinal real ou escalar para medir o valor das ações em um dado estado. Isso garante o aprendizado do agente, quando ele passa a considerar ações que podem desviá-lo ou aproximá-lo do seu objetivo (Watkins, 1989).

O sinal de reforço é uma função R arbitrária, no ambiente proposto, definida por:

$$R(s,a) \rightarrow Z \quad (7).$$

Sendo s,a ∈S,A, respectivamente.

Os valores possíveis do reforço se constituem de quatro valores inteiros, definidos como recompensa ou punição. O cálculo utilizado para a definir o sinal de reforço é dado em função de algumas condições definidas para este ambiente em específico. Elucida-se a seguir na tabela 1 as condições e valores possíveis do sinal de reforço:

Tabela 1. Definições dos sinais de reforço

Recompensa	Punição
r1 = 10	r3 = -10
r2 = 10	r4 = -10
R_{max} = 20	R_{max} = -20

Recompensa: Se o jogador tiver uma quantidade de cartas menor que o seu adversário recebe r1. Caso a diferença for igual ou maior que o valor 4; ou a quantidade própria de cartas atual é igual a 1; o agente recebe o sinal r2 como acréscimo.

Punição: Se o jogador tiver uma quantidade de cartas maior que o seu adversário, recebe r3. Caso a diferença for menor ou igual ao valor (-4); ou a quantidade de cartas atual do adversário é igual a 1; o agente recebe o sinal r4 como acréscimo.

4. O EXPERIMENTO E OS RESULTADOS

Os experimentos, enumerados a seguir, foram realizados seguindo o modelo de ambiente supracitado através da plataforma computacional desenvolvida para o fim deste trabalho. Este trabalho experimental dependeu exclusivamente do sistema de simulações citado ao permitir inúmeras simulações de partidas do jogo de cartas Uno, sendo este, retratado fielmente no ambiente computacional em relação ao modelo proposto nesta pesquisa.

Os objetivos principais dos experimentos norteiam duas metas principais deste trabalho. Uma é construir um agente artificial com aprendizado de máquina por reforço que obtenha um maior número de vitórias diante de um agente munido de inteligência básica que usa um sistema especialista para a tomada de decisão. A outra meta é validar a viabilidade de aplicação do Q-Learning em um ambiente complexo presente nos jogos de cartas.

Para validar os objetivos dos quais se propôs esta pesquisa, foram definidas cinco baterias de testes. Cada uma consistiu em executar 5.000 partidas do jogo, sendo a distribuição de cartas de cada partida dada de forma aleatória a fim de se aproximar com uma partida real do jogo Uno. Os parâmetros do Q-Learning foram definidos após a realização de diversas simulações com valores limites permitidos para cada um deles. Sendo assim, foram definidos os valores que apresentaram os melhores resultados para compor uma bateria de testes. Nota-se também que as variações nos parâmetros não causaram grandes efeitos, sendo quase desprezíveis na prática nos testes realizados com o ambiente desenvolvido. Em todas as baterias de testes, apesar da distribuição de cartas randômica, os parâmetros e critérios de aprendizado foram definidos igualmente visando realizar uma comparação mais consistente entre os testes realizados. Adiante, estabelece-se os parâmetros do algoritmo Q-Learning utilizados nas baterias de testes, bem como as devidas justificativas e os resultados obtidos.

Definiu-se o fator de aprendizagem α como elucidado na seção 2.1. O fator de desconto definido em $\gamma = 0.95$. E a taxa de exploração dada por $\varepsilon = 11$.

Após avaliar a convergência do aprendizado ao longo do tempo e mensurar a quantidade relativa de vitórias total ao longo de 5.000 simulações, os dados obtidos foram explicitados na tabela 2, que ilustra os resultados obtidos pelo agente com aprendizado com reforço:

Após avaliar a convergência do aprendizado ao longo do tempo e mensurar a quantidade relativa de vitórias total ao longo de 5.000 simulações, os dados obtidos foram explicitados na tabela 2, que ilustra os resultados obtidos pelo agente com aprendizado com reforço:

Tabela 2. Simulações de partidas do jogo

Bateria	Quantidade de vitórias	Vitórias Totais Relativas (%)
1	2618	52%
2	2690	54%
3	2669	53%
4	2745	55%
5	2652	53%
Média	2674,80	53%

O número de vitórias do agente com aprendizado por reforço se mantém em média na faixa de 53% ao longo de cinco baterias. Percebe-se uma variação mínima para o seu pior desempenho e na melhor performance, dadas, respectivamente, nas baterias 1 e 4. Isso significa que mesmo em condições aleatórias e ambíguas do ambiente, o aprendizado se manteve estável e benéfico em todos as simulações realizadas.

Evidencia-se na Figura 2, o desempenho superior do agente A sobre o agente B em relação ao número de vitórias acumuladas ao longo do tempo. Para fins práticos, denominou-se Agente A o agente que implementa o Q-Learning na tomada de decisões. O Agente B se refere ao jogador artificial que utiliza o sistema especialista para tomada de decisões.

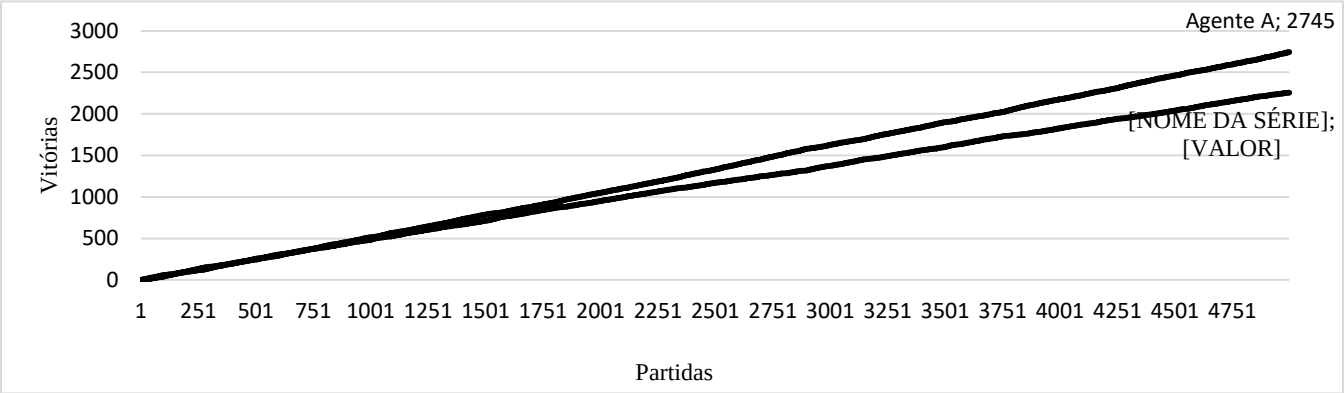

Figura 2. Descreve o desempenho acumulado dos agentes na Bateria de testes

Percebe-se também que o algoritmo oscila o aprendizado em alguns pontos devido ao problema de Perceptual Aliasing (Watkins 1989). No entanto, após 2000 partidas, o agente A consegue superar o agente B, sendo visível a estabilização linear do aprendizado.

Esperava-se que o algoritmo não encontrasse uma política ótimas de ações dadas as condições ambientes, todavia, torna-se considerável a superioridade e estabilização do aprendizado do jogador que implementa o Q-Learning perante seu adversário, levando-o a um crescimento linear de vitórias acumuladas ao longo do tempo.

5. CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho procurou identificar o desempenho em relação ao número de vitórias do agente munido com aprendizado de máquina ao utilizar um popular algoritmo de AR em ambientes estocásticos. Explicita-se adiante os desafios encontrados e as vantagens de utilização dessa aplicação em ambientes similares bem como propomos otimizar a aplicação do método com outras técnicas de IA.

No que tange aos desafios encontrados, estes se concentram na ambiguidade produzida pelo sistema na computação dos estados, produzindo um ambiente parcialmente observável, e, portanto, o Q-Learning fica passível de ruídos e oscilações na convergência do aprendizado. Mesmo assim, o agente conseguiu obter um número de vitórias maior que o agente sem aprendizado, comprovando a vantagem da utilização do AR em um ambiente caótico e complexo.

A viabilidade de implantação do método utilizado nesta pesquisa em cenários parecidos poderá ser vantajosa, já que em ambientes com um número de estados e ações reduzidos, o algoritmo parece não impactar no desempenho das aplicações. Além disso, a sua aplicação em outros jogos de cartas ou em ambientes similares poderá ser utilizada para criação de níveis de dificuldade distintos, proporcionando partidas mais desafiadoras e reais aos jogadores.

Espera-se que em pesquisa futuras, outras soluções de aprendizado possam ser integradas e um estudo matemático rigoroso seja feito em conjunto para validar formalmente a técnica empregada e otimizar eventuais problemas não retratados com maior detalhe nesta pesquisa.

Futuros experimentações poderão comparar o desempenho de agentes com outros algoritmos de aprendizado como o Sarsa e até na utilização de métodos de aprendizado por reforço baseados em modelo. Segundo Watkins (1989), tais técnicas podem garantir uma política subótima diante da informação não perfeita gerada por cenários caóticos, todavia, os gastos computacionais gerados deverão ser fortemente considerados.

6. REFERÊNCIAS

[1] BENICASA, A. X. Navegação autônoma de robôs baseada em técnicas de mapeamento e aprendizagem de máquina. Revista Brasileira de Computação Aplicada, Passo Fundo, v. 4, n. 1, p. 102-111, Março 2012.

[2] FAZIO, V. S. Algoritmos para um jogador inteligente de Poker. 2008. (Bacharelado em Ciências da Computação) - Centro Tecnológico Bacharelado, Universidade Federal De Santa Catarina, Florianópolis.Fröhlich, B. and Plate, J. 2000.

[3] FARIA, G.; ROMERO, R. F. Explorando o Potencial de Algoritmos de Aprendizado com Reforço em Robôs Móveis. In:Proceedings of the IV Brazilian Conference on Neural Networks. IV. 1999, São José dos Campos, p. 237-242, 1999.

[4] FERREIRA, J. C. L. Opponent Modelling in Texas Hold'em: Learning Pre-Flof Strategies in Multiplayer Tables. 2008. Dissertação (Mestrado em Informática) - Faculdade de Engenharia do Porto, Porto, 2008.

[5] I. BRAVI, S. LUCAS, D. PEREZ-LIEBANA, AND J. LIU. Rinascimento: Optimising Statistical Forward Planning Agents for Playing Splendor. arXiv preprint arXiv:1904.01883, 2019.

[6] KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, Journal of Arti, v. 4, p. 237-285, Maio 1996.

[7] LITTMAN, M. L. Reinforcement learning improves behaviour from evaluative feedback. Nature, v. 521, p. 445-451, Maio 2015.

[8] MENDES, P. D. D. C. High-Level Language to build Poker Agents. 2008. Dissertação (Mestrado em Informática) - Faculdade de Engenharia do Porto, , Porto, 2008.

[9] MONTEIRO, S. T.; RIBEIRO, C. H. C. DESEMPENHO DE ALGORITMOS DE APRENDIZAGEM POR REFORÇO. Controle e Automação, São José dos Campos, v. 15, n. 3, p. 320-338, Julho 2004.

[10] NETO, G. P. B. Aprendizado por reforço aplicado ao combate em jogos eletrônicos de estratégia em tempo real. 2013. 77 f. Dissertação (Mestrado em Informática) - Centrde Informática, Universidade Federal do Paraíba, João Pessoa.

[11] OTTONI, A. L. C.; NEPOMUCENO, E. G.; OLIVEIRA, M. S. D. Análise do desempenho do aprendizado por reforço na solução do problema da mochila multidimensional. Revista Brasileira de Computação Aplicada, Passo Fundo, v. 9, n. 3, p. 57-70, Outubro 2017.

[12] PENG, J.; WILLIAMS, R. J. Incremental Multi-Step Q-Learning. Machine Learning, Boston, v. 22, p. 283-290, Março 1996.

[13] PEREIRA, A. B. Q learning pessimista - um algoritmo para geração de bots de jogos em turno. 2012. 63 f. Dissertação (Mestrado em Informática)- PUC Rio, São Paulo.

[14] RUSSELL, S.; NORVIG, P. Inteligência Artificial. 3. ed. [S.l.]: Elsevier, 2013.

[15] SELVATICI, A. H. P. AAREACT: uma arquitetura comportamental adaptativa para robôs móveis que integra visão, sonares e odometria. Dissertação (Mestrado em Engenharia Elétrica)- Escola Politécnica da Universidade de São Paulo, São Paulo, São Paulo, 2005.

[16] SETHY, H.; PATEL, A.; PADMANABHAN, V. Real Time Strategy Games: A Reinforcement Learning Approach. Procedia Computer Science, v. 54, p. 257-264, Agosto 2015.

[17] SUTTON, R. S.; BARTO, A. G. Reinforcement Learning: An Introduction. 2. ed. Massachusetts : MIT Press, 2017.

[18] TESAURO, G. Temporal difference learning and td-gammon. Communications of the ACM, New York, v. 38, n. 3, p. 58-68, Março 1995.

[19] WATKINS, C. J. C. H. Learning from Delayed Rewards. [S.l.]: [s.n.], 1989.