

3 Algoritmos Implementados

Neste capítulo descreveremos os algoritmos que implementamos para Execução Otimizada de Transações.

3.1 Considerações Iniciais

Na seção 1.1, definimos o problema de execução otimizada de transações, que consiste em vender (ou comprar) um determinado volume V de ativos durante um intervalo especificado de tempo H , de maneira que o montante recebido (ou pago, no caso de compra) seja máximo (ou mínimo, no caso de compra). Nesse trabalho, implementamos estratégias de negociação que objetivam resolver o caso particular do problema de execução otimizada de transações quando $V = 1$. Para implementar algoritmos que resolvam esse problema, inserimos mais uma variável, chamada de *resolução*. A resolução, representada por R , divide o horizonte H em R pontos distintos e equidistantes que definem os instantes em que as estratégias poderão fazer alguma oferta. Por exemplo, se $H = 5$ min e $R = 10$, nossas estratégias só poderão fazer alguma oferta a cada $5/10$ minutos, ou seja, a cada 30 segundos. Todas as estratégias implementadas se submetem a essa restrição. Sendo assim, pela nossa definição, ofertas só podem ser feitas a cada H/R segundos. Suponha que $H = 10$ s e $R = 2$, por definição, as estratégias poderiam fazer ofertas nos instantes 0,5 e 10.

Nos momentos em que as estratégias podem ofertar, um tipo especial de oferta pode ser feita. Esse tipo de oferta provoca uma execução no mesmo instante, ou seja, a intenção sequer é posicionada no *book*. Para que isso seja possível, uma oferta de compra deve ter preço maior ou igual ao *ask* e uma de venda deve ter preço menor ou igual ao *bid*. Chamamos esse tipo de oferta de *oferta de execução imediata*.

Nas próximas seções mostramos todas as estratégias que implementamos para a resolução do problema com as restrições impostas.

3.2

Estratégia Submit and Execute

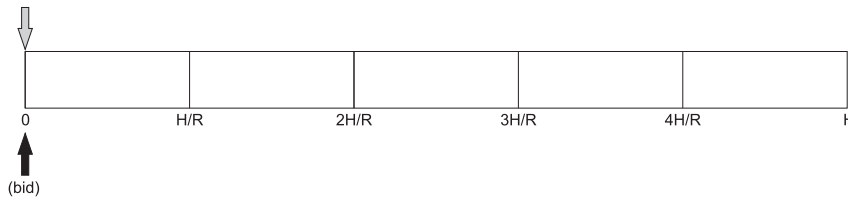


Figura 3.1: Exemplo da Estratégia *Submit and Execute* (com $R = 5$)

Na Estratégia *Submit and Execute*, apenas uma oferta é feita. No início da negociação, a estratégia faz uma oferta de execução imediata, ou seja, a oferta é executada no momento em que é feita. Na Figura 3.1, podemos ver o comportamento dessa estratégia para uma negociação em que o objetivo é vender uma ação. No exemplo da figura, dividimos o horizonte em cinco partes, ou seja, $R = 5$ (todos os exemplos seguintes do capítulo seguem o mesmo padrão do apresentado nesta seção: objetivo de venda e $R = 5$). A seta mais escura significa uma oferta, a mais clara marca o instante da execução da última oferta feita e o valor entre parênteses abaixo da seta escura representa o valor da oferta. Assim, nesse exemplo, no instante zero a estratégia faz uma oferta de venda com o valor igual ao *bid*, ou seja, vende a ação para quem está pagando mais no momento. Se fosse uma negociação de compra, a oferta teria valor igual ao *ask*.

Dado que a estratégia sempre ajusta o preço da oferta de acordo com o preço do *book* oposto (paga o *ask* na compra e aceita o *bid* na venda), o período de negociação da estratégia nunca passa do instante zero.

3.3 Estratégia Submit and Leave



Figura 3.2: Estratégia *Submit and Leave*: Duas Ofertas (com $R = 5$)

Na Estratégia *Submit and Leave*, uma oferta inicial é feita com valor igual ao *ask* no caso de venda (respectivamente, com valor igual ao *bid*). A estratégia aguarda a execução da oferta durante todo o período de negociações. Caso a oferta seja executada, a estratégia encerra sua execução. Se chegarmos

ao instante H com a oferta ainda não executada, retiramos a oferta existente do mercado e fazemos outra de execução imediata.

Na Figura 3.2, podemos ver um exemplo de execução da estratégia. No instante zero uma oferta é feita e sua execução é aguardada, como não ocorre, no instante H , a estratégia retira a oferta anterior e faz outra de execução imediata.

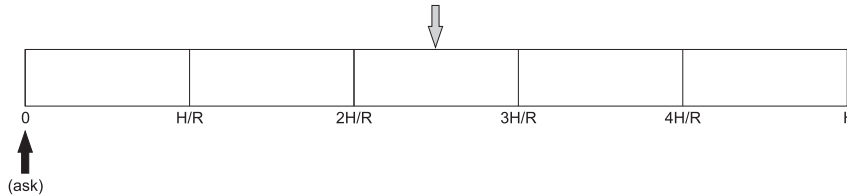


Figura 3.3: Estratégia *Submit and Leave*: Única Oferta (com $R = 5$)

Na Figura 3.3, podemos ver um exemplo diferente de execução da estratégia *submit and leave*. Como na Figura 3.2, uma oferta é feita logo no instante zero, a diferença é que, logo após ao instante $2H/R$, a oferta feita é executada e a estratégia encerra sua execução.

3.4

Estratégia Reinforcement Learning

Essa estratégia, proposta por Nevmyvaka (Nevmyvaka et al. 2006), faz uso de aprendizado por reforço para gerar um política de negociação. Na seção 3.4.1, formulamos o problema no modelo de aprendizado por reforço, na seção 3.4.2, detalhamos o algoritmo de aprendizado implementado e por último, na seção 3.4.3, apresentamos um exemplo de execução da estratégia.

3.4.1

Formulação

Nesta seção, formulamos o problema de execução otimizada de transações no modelo de aprendizado por reforço, para que possamos executar o algoritmo *value iteration* (seção 2.2.5) com o objetivo de encontrar uma política de negociação ótima. Definimos a seguir os estados, ações e recompensas.

Estados

Cada estado $s \in S$ é um vetor de atributos (variáveis de estado) que descreve a atual configuração do nosso ambiente. Em ambientes complexos como os mercados financeiros, representar todos os aspectos relevantes como um vetor de atributos é uma tarefa nada trivial. Por isso, no nosso trabalho, quando nos referimos ao *estado* do mercado, estamos nos referindo ao *estado*

observado, ou seja, estamos fazendo referência à porção do mercado que conseguimos representar.

Dividimos os atributos em dois grupos: públicos e privados. Os atributos privados são aqueles que definem o estado atual da estratégia e os públicos expressam a atual configuração do mercado. A lista de todos os atributos privados pode ser vista na Tabela 3.1. O atributo t informa para o agente de aprendizado em qual ponto do tempo ele se encontra no período de negociação. O e informa se as negociações estão encerradas ($e = 1$) ou não ($e = 0$). As negociações se encerram quando uma oferta feita pela estratégia é executada. Por último, o atributo f informa se a estratégia tem alguma oferta feita no mercado ($f = 1$) ou não ($f = 0$).

Atributo	Descrição	Possíveis Valores
t	Tempo de negociação	$\{0, H/R, 2H/R, \dots, (R-1)H/R, H\}$
e	Negociações Encerradas	$\{0, 1\}$
f	Oferta feita	$\{0, 1\}$

Tabela 3.1: Atributos Privados

Os atributos públicos, que informam ao agente a configuração atual do mercado, podem ser vistos na tabela 3.2. O atributo q dá ao agente informações sobre o volume de ativos que está localizado no topo de cada *book*. Seja V_{compra} o volume de ativos nas ofertas de compra no topo do *book* de compra e V_{venda} o volume de ativos que está a venda no topo do *book* de venda. Se $V_{compra} \geq (1 + \lambda)V_{venda}$, então $q = 1$. Se $V_{venda} \geq (1 + \lambda)V_{compra}$, então $q = 2$ e se $V_{compra} < (1 + \lambda)V_{venda}$ e $V_{venda} < (1 + \lambda)V_{compra}$, então $q = 0$. O parâmetro λ define uma margem de folga na comparação entre os volumes. Por exemplo, se $\lambda = 0.2$ e $q = 1$, significa que V_{compra} é no mínimo 20% maior que V_{venda} . Se a diferença for menor que 20%, então $q = 0$. Sendo assim, o atributo q informa ao agente sobre uma possível pressão existente nas filas, ou seja, se há mais ações sendo vendidas que compradas, se há mais ações sendo compradas que vendidas ou se há uma pressão indefinida entre os volumes negociados.

Atributo	Parâmetro	Descrição	Possíveis Valores
q	Tolerância (λ)	Pressão das filas	$\{0, 1, 2\}$
c	Tamanho (k)	Último contexto	$c = (v_1, v_2, \dots, v_k)$ $\forall v_i \in \{s, l, h\}, 0 \leq i \leq k$

Tabela 3.2: Atributos Públicos

O atributo c é um vetor com k componentes que dá ao agente informações sobre as últimas flutuações no valor de negociação da ação que se deseja negociar. Para a construção do vetor c , que será inserido na representação

do estado, analisamos as k últimas variações do valor de negociação da ação em questão. Cada componente v_i do vetor c pertence ao conjunto $\{s, l, h\}$, que significa que o valor de negociação da ação permaneceu estável, desceu ou subiu, respectivamente. Por exemplo, suponha que $k = 3$, que os valores de negociação da ação flutuaram segundo o gráfico da Figura 3.4 e que estamos no instante $t = 10$. Para a construção do valor de $c = (v_1, v_2, v_3)$ no instante 10, iniciamos analisando do instante 7. Do instante 7 para o 8 o valor da ação subiu de R\$ 21,12 para R\$ 21,13, assim $v_1 = h$, do 8 para o 9 o valor da ação desceu, fazendo com que $v_2 = l$ e por último, do instante 9 para o 10, o valor permaneceu em R\$ 21,12, implicando em $v_3 = s$. Por essa análise, podemos concluir que c no instante 10 para $k = 3$ é $c = (h, l, s)$.

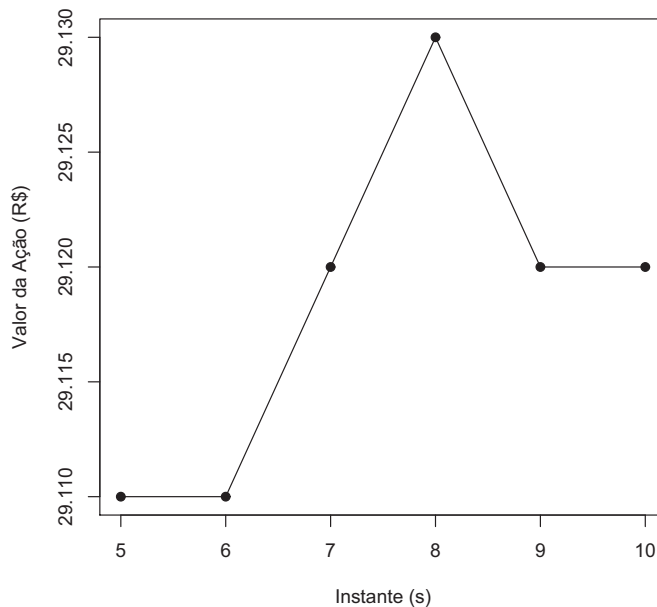


Figura 3.4: Exemplo de Flutuação do Valor de Negociação de uma Ação

Assim, nossa representação de estado é composta por cinco atributos. Todo estado $s \in S$ é um vetor de cinco componentes $s = (t, e, f, q, c)$ onde t , e e f armazenam informações sobre o estado da estratégia e q e c informam ao agente sobre a atual configuração do mercado.

Ações

Baseada no estado atual do ambiente, a estratégia deverá fazer uma ação a partir de uma política de negociação. Definimos um conjunto de ações que poderão ser feitas. Todas as possíveis ações podem ser vistas na tabela 3.3. A tabela possui quatro colunas, as quais são detalhadas abaixo.

Nome	Compra	Venda	Requisitos
$A_{1,i}$	$bid - i$	$ask + i$	$e = 0$ e $t < H$
A_2	ask	bid	$e = 0$
A_3	faz nada	faz nada	$e = 0$ e $t < H$
A_4	retira oferta	retira oferta	$e = 0$ e $f = 1$ e $t < H$
A_5	$(bid + ask)/2$	$(bid + ask)/2$	$e = 0$ e $t < H$

Tabela 3.3: Espaço de Ações do Agente

- Primeira coluna: nome da ação;
- Segunda coluna: preço da oferta que é feita pela ação no caso de uma negociação de compra;
- Terceira coluna: preço da oferta no caso de venda;
- Quarta coluna: requisitos que devem ser atendidos para que seja possível a realização da ação.

Pela tabela, podemos ver que se estivermos em algum instante antes de H em que as negociações não tenham sido encerradas e a estratégia visa vender, a ação $A_{1,3}$ faz uma oferta com preço três centavos maior que o ask (todas as somas e subtrações com um escalar apresentadas na tabela se referem a centavos). A primeira linha da tabela, a das ações $A_{1,i}$, representa todas as ações que fazem ofertas. Durante a execução do algoritmo de aprendizado por reforço, definimos quarenta e uma ações possíveis derivadas de $A_{1,i}$, para isso, variamos o parâmetro i de -20 a 20 , ou seja, durante o aprendizado $i \in \{-20, -19, \dots, 19, 20\}$. Podemos ver também que se chegarmos ao instante H com as negociações ainda não encerradas, apenas a ação A_2 é possível, que faz uma oferta de execução imediata. Existem duas ações especiais que não fazem nenhuma oferta: A_3 e A_4 . Como a ação A_4 exige que alguma oferta tenha sido feita pela estratégia (requisito $f = 1$), sempre que ela for feita, a oferta anteriormente submetida é retirada do mercado. A ação A_3 não faz nenhuma oferta nem retira uma existente, apenas faz a estratégia aguardar o próximo momento para fazer outra ação. Vale salientar que a execução da ação A_4 está atrelada a cada uma das ações que geram alguma oferta no mercado ($A_{1,i}$, A_2 e A_5), o que significa que, cada vez que o agente for submeter uma oferta, uma possível oferta anteriormente submetida é retirada do mercado, o que obriga ao agente ter no mercado apenas uma oferta de cada vez.

Pelo conjunto de ações que definimos, a estratégia nunca poderá sair de um período de negociações sem nenhuma oferta executada. A ação A_2 obriga a estratégia a executar imediatamente uma oferta caso chegue ao último instante de negociação (H) com $e = 0$. A estratégia se sujeita a maior oferta

de compra (*bid*) para vender o seu ativo ou a menor oferta de venda (*ask*) para comprar um.

Estado	Possíveis Ações
$s_1 = \{t = 5, e = 1, f = 0, q = 0, c = (s, l, s)\}$	$\{\emptyset\}$
$s_2 = \{t = H, e = 0, f = 0, q = 2, c = (l, l, s)\}$	$\{A_2\}$
$s_3 = \{t = 0, e = 0, f = 1, q = 1, c = (s, s, s)\}$	A
$s_4 = \{t = 8, e = 0, f = 0, q = 0, c = (h, h, h)\}$	$A - \{A_4\}$

Tabela 3.4: Exemplo de Possíveis Ações em Estados

A Tabela 3.4 exibe alguns exemplos das possíveis ações que o agente poderia fazer em alguns estados diferentes, onde $A = \{A_{1,i}, A_2, A_3, A_4, A_5\}$. Em s_1 , como $e = 1$, nenhuma ação pode ser feita, já que para todas as ações é requisito que $e = 0$. No estado s_2 , como $t = H$, apenas A_2 pode ser feita, pois apenas ela é viável quando estamos no último momento disponível para fazer uma oferta. Em s_3 , todas as ações são possíveis, pois $e = 0$ e $t < H$. No último exemplo, em s_4 , não é permitido fazer A_4 , já que $f = 0$.

Recompensas

A execução de uma ação em um determinado estado pode produzir uma recompensa imediata, que representa a quantidade de dinheiro que foi paga ou recebida, dependendo se a estratégia está comprando ou vendendo, respectivamente. Embora estejamos interessados em comparar montantes de dinheiro, não podemos exprimir os resultados em reais. Por exemplo, se uma ação, em um determinado momento, era negociada a cinco reais e em outro, a trinta, economizar dez centavos tem implicações bem diferentes em cada momento. Para podermos comparar estratégias diferentes que negociam ativos diferentes, que podem ter preços bastante desiguais, definimos o valor de uma execução de uma oferta como sendo o preço da oferta executada dividido pelo *mid-spread price* $((ask + bid)/2)$ no início do período de negociações em questão.

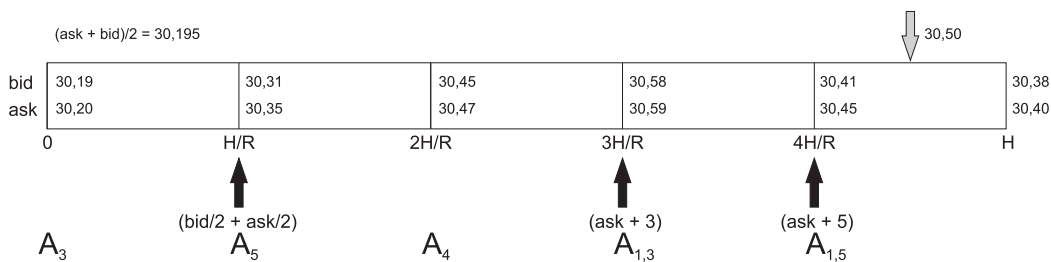


Figura 3.5: Exemplo de Recompensa

Por exemplo, na Figura 3.5 podemos ver um período de negociações de venda onde a estratégia faz uma ação em cada ponto que é permitido. No início do período, a estratégia faz a ação A_3 , que não provoca nenhuma oferta e por isso a estratégia não recebe nenhuma recompensa por ela ($r = 0$), no instante H/R a estratégia faz a ação A_5 , que faz com que uma oferta de preço $(ask + bid)/2$ seja submetida ao mercado. Como a oferta não é executada até o instante $2H/R$, o sinal de reforço recebido também é zero. No instante $2H/R$, a ação A_4 é feita, fazendo com que a oferta seja retirada do mercado e nenhum sinal de reforço seja recebido. No instante $3H/R$ a estratégia faz uma oferta com preço $(ask + 3) = 30,62$, mas pelo mesmo motivo de quando a ação A_5 foi feita, nenhuma recompensa é recebida. Quando chegamos ao instante $4H/R$, a estratégia faz a ação $A_{1,5}$, que faz com que uma oferta de preço 30,50 seja feita. Em algum momento após $4H/R$, a oferta é executada. A execução da oferta gerou uma receita de 30,50, como o *mid-spread price* no início do período de negociações era 30,195, a recompensa recebida pela estratégia é de $30,50/30,195 = 1,010$.

3.4.2

Algoritmo de Treinamento

O algoritmo que usamos para encontrar uma política de negociações é uma adaptação do *value iteration* apresentado na seção 2.2.5. O objetivo do nosso algoritmo é encontrar a política ótima π que mapeia estados em ações, de maneira que a soma das recompensas seja máxima no caso de venda e mínima no caso da compra. Para o treinamento do algoritmo, contamos com um histórico dos valores das ações da Petrobras coletado durante dois meses. Dividimos esses dados em vários períodos de negociação. Chamamos de P o conjunto de todos os períodos de negociação encontrados. O pseudocódigo do algoritmo implementado para encontrar a melhor política de negociações é apresentado a seguir.

O algoritmo inicia nossa função valor com zero para todos os possíveis valores de s (linha 1), depois disto, inicia a iteração sobre todos os estados. O algoritmo inicia dos últimos estados até os primeiros em relação ao tempo (*backward induction*), por exemplo, o primeiro estado a ser aprendido será um onde $t = H$, depois onde $t = (R - 1)H/R$, e assim segue até o último estado, quando $t = 0$. Isso nos permite saber em cada estado, durante o aprendizado, qual o valor de V^π para o estado seguinte, por exemplo, se estamos em um estado onde $t = 0$, pela maneira que a iteração é feita, já aprendemos qual o valor de V^π para todos os estados onde $t = H/R$. Para cada $s \in S$, fazemos uma iteração em todos os períodos de negociação do conjunto de trei-

Algoritmo 3 *Value Iteration* Adaptado

```

1:  $V^\pi(s) = 0, \forall s \in S$ 
2:  $n(s, a) = 0, \forall (s, a) \in S \times A$ 
3: for all  $s \in S$  do
4:   for all  $periodo \in P$  do
5:     if  $valido(s, periodo)$  then
6:       for all  $a \in A(s)$  do
7:          $r \leftarrow R_{im}(s, a)$ 
8:          $Q^\pi(s, a) \leftarrow (n(s, a) \cdot Q^\pi(s, a) + r + V^\pi(s')) / (n(s, a) + 1)$ 
9:          $n(s, a) \leftarrow n(s, a) + 1$ 
10:      end for
11:    end if
12:  end for
13:   $V^\pi(s) \leftarrow melhor(Q^\pi(s, a))$ 
14:   $\pi(s) \leftarrow a$ 
15: end for

```

namento (linha 4). Antes de iniciar os testes com todas as ações, verificamos se o estado é válido no período em questão (linha 5), isto é, checamos se os valores de todos os atributos do estado atual são viáveis no período atual, por exemplo, se $s = \{t = H, e = 0, f = 0, q = 2, c = (l, l, s)\}$ e o estado lido do período atual no instante $t = H$ for $s_{lido} = \{t = H, e = 0, f = 0, q = 0, c = (l, l, s)\}$, o estado não é válido, já que em s , $q = 2$, e em s_{lido} , $q = 0$. Para que o algoritmo inicie o cálculo de Q^π para um estado em um determinado período, é obrigatório que $s = s_{lido}$. Se o estado for válido, testamos todas as ações possíveis nele (linha 6). Na linha 7, armazenamos em r a recompensa que será recebida se fizermos a ação a em s , depois disto, na linha 8, atualizamos o valor de Q^π , onde $n(s, a)$ é o número de vezes que tentamos a em s e s' é o estado para qual o ambiente vai se fizermos a ação a em s . Depois de calcularmos Q^π para todas as ações em todos os períodos para s , atualizamos o valor de $V^\pi(s)$ com o melhor valor de $Q^\pi(s, a)$ (no caso de compra, o valor é o mínimo e no caso de venda, o máximo). Por último, antes de iniciarmos o aprendizado do próximo estado, atualizamos a política $\pi(s)$ com a ação que provoca o melhor valor de $Q^\pi(s, a)$ (linha 13).

O Algoritmo 3 faz uma iteração em todos os estados, e para cada iteração faz outra em todos os períodos até fazer outra em cada ação possível, assim, a complexidade do algoritmo é $O(|A| \cdot |S| \cdot |P|)$.

3.4.3**Exemplo**

Seja π a política ótima de negociações de venda aprendida pelo Algoritmo 3 apresentada na Tabela 3.5 para um determinado conjunto de treinamento.

A Tabela 3.5 mostra alguns mapeamentos feitos pela política para guiar a estratégia durante o período de negociações. A primeira coluna especifica alguns estados e a segunda mostra a melhor ação aprendida para o estado.

Estados	$\pi(s)$
$s_1 = \{t = 0, e = 0, f = 0, q = 0, c = (s, s, s)\}$	A_3
$s_2 = \{t = H/R, e = 0, f = 0, q = 1, c = (s, l, l)\}$	A_5
$s_3 = \{t = 2H/R, e = 0, f = 1, q = 0, c = (l, l, s)\}$	A_4
$s_4 = \{t = 3H/R, e = 0, f = 0, q = 2, c = (s, s, s)\}$	$A_{1,3}$
$s_5 = \{t = 4H/R, e = 0, f = 1, q = 0, c = (s, l, s)\}$	$A_{1,5}$

Tabela 3.5: Exemplo de Política de Venda

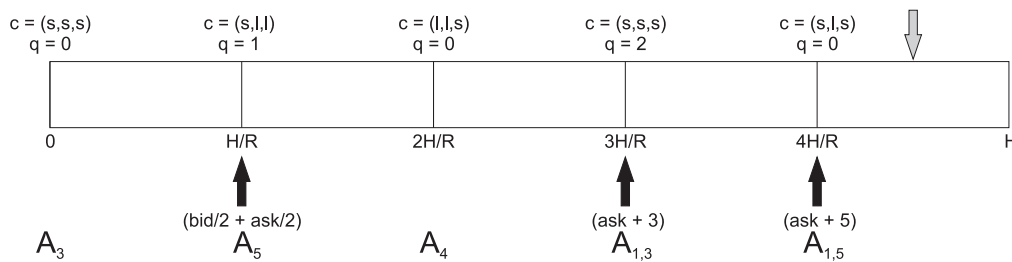


Figura 3.6: Exemplo da Estratégia *Reinforcement Learning*

Na Figura 3.6 podemos ver um período de negociações de venda. Nesse exemplo, a estratégia está sendo guiada pela política apresentada na Tabela 3.5. No instante 0, o estado atual do ambiente (período de negociações) é $\{t = 0, e = 0, f = 0, q = 0, c = (s, s, s)\}$, então a estratégia consulta a política para saber qual a melhor ação para o estado. A política retorna a ação A_3 e a estratégia faz a mesma. No instante H/R , o estado atual é $\{t = H/R, e = 0, f = 0, q = 1, c = (s, l, l)\}$ e a ação A_5 é feita. A política é seguida até que chegamos no instante $4H/R$, onde o estado do ambiente é $\{t = 4H/R, e=0, f=1, q=0, c=(s, l, s)\}$, por isso, a ação $A_{1,5}$ é feita e a ação é vendida antes de chegarmos ao instante H , encerrando a participação da estratégia.

3.5 Estratégia Kearns

A *Estratégia Kearns*, proposta por Nevmyvaka (Nevmyvaka et al. 2005), faz uso de uma metodologia analítica que visa a execução eficiente em mercados eletrônicos modernos. Essa metodologia consiste da criação e análise de três curvas para a seleção de possíveis preços eficientes de ofertas, resultando na geração de possíveis políticas de negociação.

Na seções seguintes explicamos a metodologia analítica proposta por Nevmyvaka e o algoritmo de análise utilizado para a geração de políticas de negociação em duas metodologias: básica e estendida.

3.5.1

Metodologia Básica

Nesta seção, apresentamos a metodologia analítica básica que utilizamos para a geração de políticas de negociação utilizadas pela *Estratégia Kearns*. Essa metodologia consiste da criação de duas curvas: curva de retornos esperados, que determina quais preços resultam em execuções eficientes, e curva de risco, que determina o risco de não execução para cada preço de oferta. Combinando essas duas funções, podemos extrair a *fronteira eficiente de preços*, que fornece para a estratégia um conjunto de retornos esperados em função do risco de não execução. A seguir apresentamos as curvas e os algoritmos usados para a geração de cada.

Preço Esperado de Execução

Usamos uma estratégia simples para guiar as negociações. No início do período de negociações, submetemos uma oferta limitada com um determinado preço e esperamos. Se no último instante do período de negociações a oferta ainda não tiver sido executada, ela é retirada do mercado e uma oferta de execução imediata é feita. Essa estratégia de negociação é idêntica a *Submit and Leave*. O objetivo da metodologia básica é identificar o preço inicial da estratégia *Submit and Leave* que implicará no lucro máximo.

É importante definir primeiro o que exatamente estamos medindo. Pelo mesmo motivo que foi exposto na seção 3.4.1 (grandes variações entre preços de ativos), o valor que desejamos otimizar deve sofrer uma normalização. Portanto, a variável que buscaremos otimizar (retornos) é a diferença entre o preço de execução e o *mid spread price* do início do período de negociações, expressada como fração do *mid spread price* inicial. Por exemplo, digamos que, no início de um período, o *bid* seja R\$ 24.12 e o *ask* R\$ 24.18, então o *mid spread price* inicial será de R\$ 24.15. Se uma oferta de compra for executada a R\$ 24.18, nosso preço diferencial será $(24.15 - 24.18)/24.15 = -0.0012$, se fosse uma oferta de venda, o preço diferencial seria $(24.18 - 24.15)/24.15 = 0.0012$. Expressaremos esses números em pontos base (*basis points*), que seria *12bps* nesse exemplo.

Em cada período de negociação, submetemos ofertas iniciais com vários preços diferentes (definidos pelo parâmetro *fator* ad estratégia *Submit and Leave*) e armazenamos a média dos retornos para cada um. Esses retornos tendem a ser altos em um certo intervalo de preços, que representa os preços que alcançam os preços de execução mais vantajosos. Uma curva de preço-retorno pode ser vista na Figura 3.7.

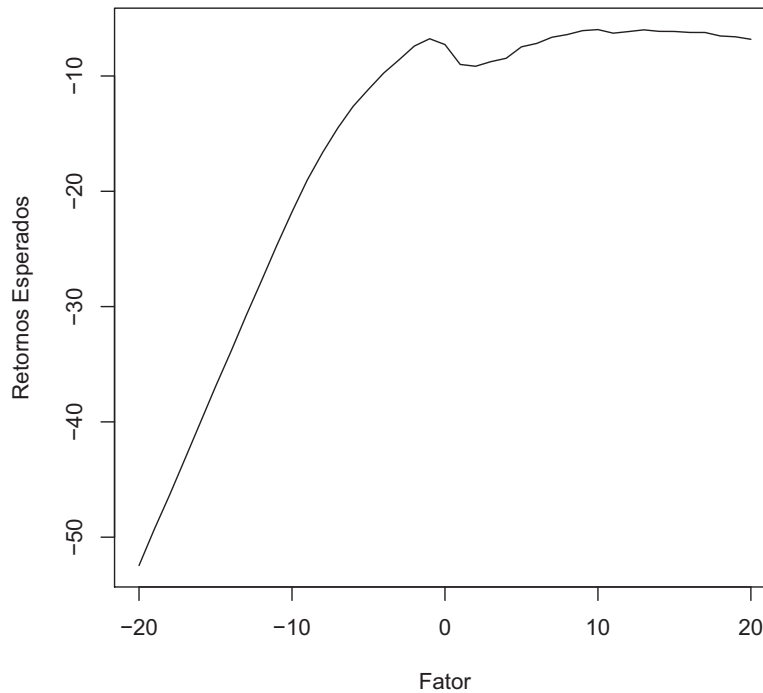


Figura 3.7: Curva de Retornos

No eixo x , plotamos os fatores que definem os preços iniciais da estratégia *Submit and Leave*. Os preços apresentados no gráfico são relativos ao preço do topo do *book* da oferta. Por exemplo, se a oferta for de compra e $x = 10$, significa que a oferta inicial possui preço igual a $bid - 10 \cdot inc$, se $x = -10$, o preço será de $bid + 10 \cdot inc$, onde inc representa o incremento no valor da oferta. Se $inc = 0.01$, a oferta será incrementada de centavos. No eixo y , plotamos a média dos retornos recebidos para cada preço. Olhando para a curva podemos ver que variar o preço da oferta inicial nos dá a possibilidade de receber melhores retornos. O retorno máximo da curva representa o menor custo possível de transação e o preço correspondente, marcado no eixo x , é o preço inicial ótimo.

Riscos

O risco de não execução é o segundo ponto de análise da metodologia proposta por Kearns. A curva de retornos não nos dá informações suficientes para a construção da estratégia de negociação. A estratégia de negociação poderia se basear só no preço inicial de oferta que provoca o retorno máximo, mas é válido salientar que maiores retornos vêm com maiores riscos. No nosso caso, o risco de não execução da oferta no preço inicial é que é levado

em consideração. Se uma oferta não for executada no preço inicial, seremos obrigados a nos submeter, provavelmente, a preços menos favoráveis no fim do período de negociações.

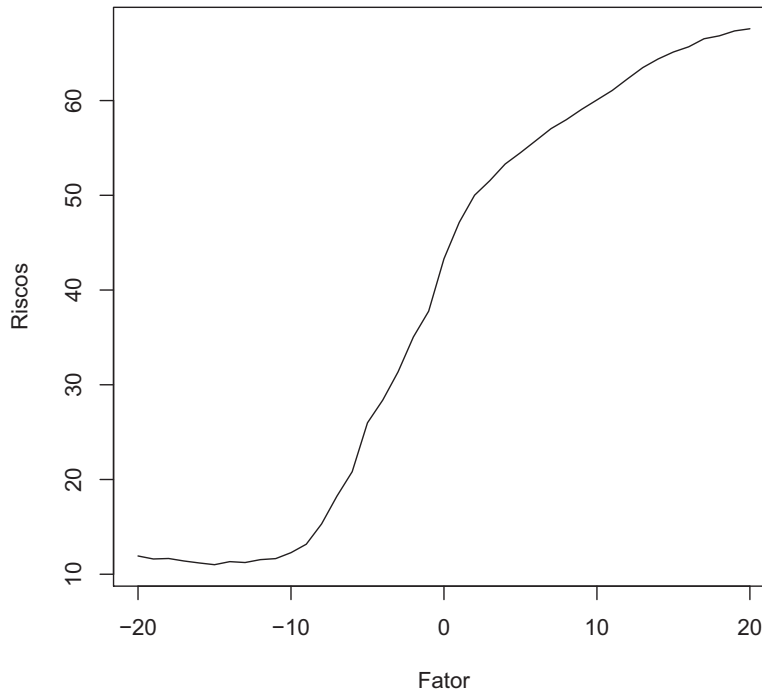


Figura 3.8: Curva de Riscos

Uma curva de risco correspondente à curva de retornos da Figura 3.7 pode ser vista na Figura 3.8. Para a criação da curva de riscos, simplesmente plotamos o desvio padrão dos retornos (eixo y) para todo possível fator que representa os preços iniciais (eixo x).

Segundo Nevmyvaka (Nevmyvaka et al. 2005), a curva de riscos geralmente é crescente, ou seja, os riscos de não execução aumentam com a diminuição do preço da oferta inicial. Quanto mais fundo no *book* uma oferta estiver, mais incerteza em relação ao seu preço final de execução existe.

Fronteira Eficiente de Preços

Depois que encontramos os perfis de retornos e riscos, precisamos de um método para combinar as duas medidas de maneira a otimizá-las juntas para derivar o preço ótimo da oferta inicial. Para plotar a curva de *riscos-retornos*, colocamos todos os possíveis preços iniciais em um gráfico bidimensional, onde o eixo x representa o risco de não execução e o eixo y seu retorno. Um

pseudocódigo que descreve a construção da curva *riscos-retornos* pode ser visto abaixo.

Algoritmo 4 Algoritmo para geração da Fronteira Eficiente de Preços

```

1: for all  $f \in F$  do
2:    $x \leftarrow getRisco(f)$ 
3:    $y \leftarrow getRetornoEsperado(f)$ 
4:    $riscosRetornos \leftarrow addPoint(x, y)$ 
5: end for

```

O algoritmo inicia sua execução fazendo uma iteração sobre todos os fatores utilizados para a criação das curvas de retornos esperados e riscos. Para cada fator, o algoritmo consulta as curvas anteriores para saber qual o retorno esperado e o risco calculado para ele (linhas 2 e 3). O ponto resultante das consultas é adicionado à curva *riscos-retornos* (linha 4). Assim, ao término da execução do algoritmo, temos uma curva que relaciona os riscos e os retornos esperados para cada um dos fatores. A curva que combina as duas anteriores pode ser vista na Figura 3.9.

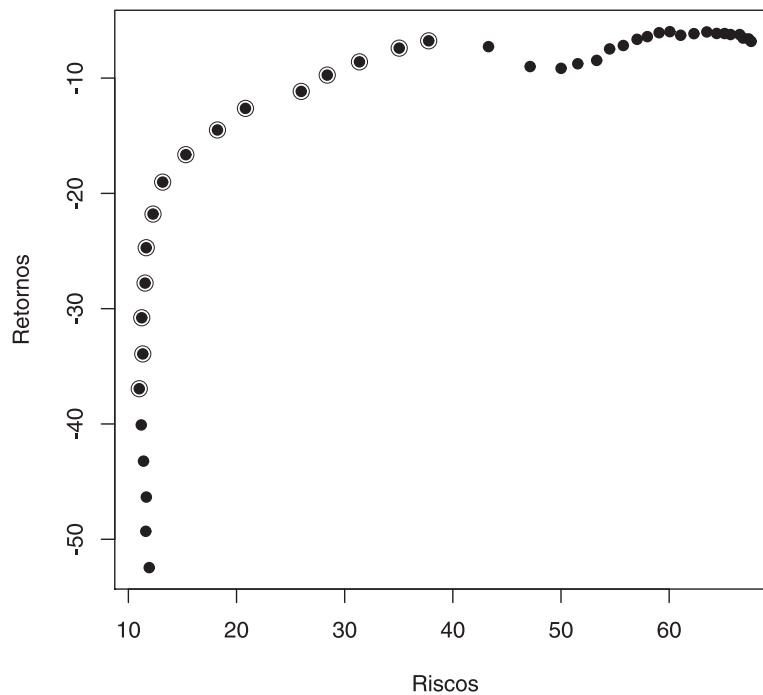


Figura 3.9: *Trade-off* entre risco e retorno

Só a parte do topo da curva, onde o aumento no risco resulta em retornos esperados maiores, deve ser usada para selecionar o preço da oferta inicial. Este trecho da curva é chamado de “Fronteira Eficiente de Preços”. Essa fronteira eficiente foi primeiramente descrita por Markowitz (Markowitz 1952). Em seu trabalho, Markowitz apresenta uma representação gráfica de uma curva onde um investidor pode consultar os preços que fornecerão um determinado

retorno em um nível mínimo de risco. Da mesma forma, o investidor pode selecionar preços com risco mínimo para um determinado retorno. Essa curva, cuja representação gráfica é utilizada pelo investidor, é denominada *Fronteira Eficiente*. Neste exemplo, a fronteira aparece no topo esquerdo da curva (pontos com círculos ao redor), conectando o ponto de mínimo risco com o de retorno máximo. Em outros pontos da curva, sempre podemos encontrar menos risco para o mesmo retorno ou melhores retornos para o mesmo risco, o que não faz sentido.

A análise proposta por Nevmyvaka (Nevmyvaka et al. 2005) é feita usando históricos dos valores das ações para executar simulações. As simulações são usadas para a construção das três curvas descritas anteriormente para que seja possível construir a fronteira eficiente de preços. Usando essa fronteira, o negociador pode escolher entre alguns preços iniciais de oferta, assumindo maiores riscos para maiores retornos ou menores riscos para menores retornos. Introduzimos um parâmetro para a seleção do preço inicial chamado de *agressividade*. Quando maior a *agressividade*, maiores riscos o negociador assume, ou seja, quanto maior a *agressividade*, mais próximo do ponto mais a direita da fronteira o negociador ofertará.

3.5.2 Metodologia Estendida

De acordo com a metodologia básica, uma vez que a oferta foi feita, no início do período de negociações, não a atualizamos até o último segundo de negociação. A ideia da metodologia estendida, proposta por Nevmyvaka (Nevmyvaka et al. 2005), é fazer várias análises nos dados para definir uma fronteira eficiente de preços para determinados instantes do período de negociações.

Assim como na *Estratégia Reinforcement Learning*, a *Estratégia Kearns* pode fazer ofertas nos pontos marcados pela resolução. Para que isto seja possível, precisamos executar uma análise dos dados (como o explicado na seção de Metodologia Básica) para cada ponto no tempo de possibilidade de oferta. Assim, a cada H/R segundos, poderemos submeter novamente a nossa oferta ainda não executada com preço atualizado.

3.6 Estratégia Aleatória

A estratégia aleatória decide por sorteio qual ação fazer a cada momento. Inicialmente, um sorteio é feito para decidir se alguma oferta será feita ou não. A probabilidade de fazer uma oferta é Pr_o e a de não fazer é $(1 - Pr_o)$. Se

a opção de fazer uma oferta for selecionada, outro sorteio é feito para saber qual ação será executada. Cada ação A_i da Tabela 3.3, que provoca oferta, tem probabilidade Pr_i de ser executada. Se a opção de não fazer oferta for escolhida, a estratégia escolhe entre A_3 (com probabilidade Pr_3) e A_4 (com probabilidade Pr_4) e executa a ação. Só existe uma possibilidade de não acontecer nenhum sorteio, que ocorre quando a estratégia chega ao instante H sem negociar sua ação, nesse caso, a ação A_2 obrigatoriamente é feita, provocando execução imediata.

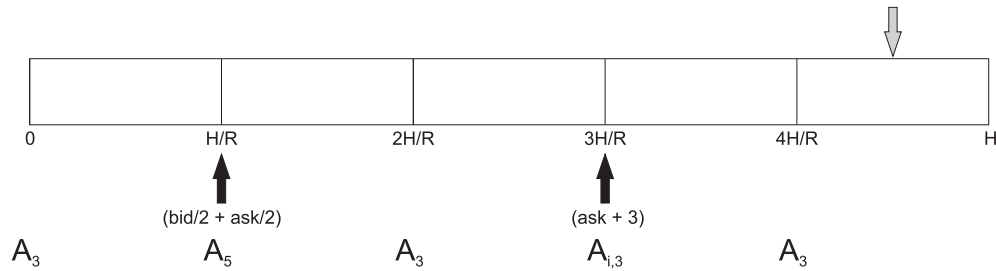


Figura 3.10: Exemplo da Estratégia Aleatória

Na Figura 3.10, podemos ver uma sequência de execução de ações da estratégia. No instante 0 foi escolhido não fazer oferta e a ação A_3 foi sorteada, no instante H/R a opção de fazer uma oferta foi escolhida e a ação A_5 foi selecionada. Os sorteios se sucedem até que em algum instante após $4H/R$ a oferta feita em $3H/R$ é executada e a estratégia se encerra.

3.7 Estratégia Ótima

A estratégia ótima não possui nenhuma restrição de execução, podendo fazer ofertas a qualquer instante (não só nos pontos marcados por R). É permitido também para esta estratégia examinar o que vai acontecer no futuro com os valores das ações. Isto possibilita sempre comprar a ação no momento que ela estiver mais barata e vender quando estiver mais cara. Apesar dessa estratégia não poder ser aplicada nos mercados reais por precisar prever o futuro, ela é interessante para efeitos de comparação. Com essa estratégia implementada, podemos ter uma ideia de quanto as outras estratégias são boas ou ruins.