4
Análise Experimental

Neste capítulo, além de explicar nossos dados, reportamos os experimentos executados a fim de comparar as estratégias apresentadas no capítulo 3.

4.1
Ambiente e Dados

Nesta seção, descrevemos o ambiente de execução dos experimentos, os dados utilizados para a simulação do mercado financeiro e o pré-processamento que foi necessário aplicar nos dados coletados.

4.1.1
Descrição do Ambiente de Execução

Os experimentos foram executados em ambiente Windows em uma máquina Intel Pentinum Dual-Core 1,46 GHz com 2,5 GB de memória RAM. Todos os algoritmos foram implementados usando linguagem Java versão 1.6.

4.1.2
Dados

Para a execução dos experimentos, contamos com dados coletados entre maio e outubro de 2008. Os dados consistem do histórico dos valores a cada segundo dos ativos da Petrobras (setembro e outubro), Vale (setembro e outubro) e Dólar (maio, junho, julho e agosto). Cada histórico é armazenado em um arquivo texto cujas linhas possuem os seguintes campos:

- **Data e Hora**: Data e hora dos valores da linha;
- ***Bid***: valor do *bid* do ativo;
- ***Ask***: valor do *ask* do ativo;
- ***Last***: preço da última oferta executada do ativo;
- **Volume Compra**: soma dos volumes das ofertas de compra de preço igual ao *bid*;
- **Volume Venda**: soma dos volumes das ofertas de venda de preço igual ao *ask*;

Como contamos apenas com os valores do *bid* e do *ask* a cada segundo, não podemos simular o funcionamento dos *books* apresentado na seção 2.1. Por isso, usamos uma abordagem diferente para saber quando uma oferta pode ser executada ou não. Suponha que uma oferta foi feita com preço igual a p (como o descrito na seção 1.1, nossas ofertas sempre tentam comprar ou vender o lote mínimo, $V = 1$). No caso de compra, nosso sistema considera que a oferta foi executada no instante em que $p \geq ask$, e no caso da venda, quando $p \leq bid$. Por exemplo, na Figura 4.1, podemos ver os valores do *bid* (linha sólida) e do *ask* (linha pontilhada) dos ativos da Petrobras em um período de 9 segundos. Suponha que uma oferta de compra é feita no instante 2 com preço igual ao *bid*, ou seja, 29.27. De acordo com a nossa metodologia, a oferta é executada no instante 5, quando o *ask* atinge o valor da oferta. De maneira semelhante, se uma oferta de venda for feita no instante 7 com valor 29.25, sua execução ocorre no instante 8, quando o *bid* atinge esse valor.

Nossas condições de execução de oferta ($p \leq bid$ ou $p \geq ask$) não refletem os mercados reais. Em uma bolsa de valores, uma oferta de compra pode ter o valor do *ask* e não ser executada (seção 2.1), pois pode ser que alguma oferta com o mesmo preço da nossa tenha sido ofertada antes, tendo assim maior prioridade de execução. Por exemplo, usando os *books* descritos nas Tabelas 2.1 e 2.2, fazemos uma oferta de compra de 100 ações com o valor de R$ 29,32, assim, nossa oferta será posiciona na segunda colocação do *book* de compra. Suponha que no decorrer das negociações, o *ask* seja de R$ 29,32, mesmo nossa oferta tendo valor igual ao *ask* (R$ 29,32) ela não será executada, pois existe uma oferta com o mesmo valor e com maior prioridade. Na nossa abordagem isso não é levado em consideração. Essa limitação da nossa abordagem se deve ao fato de não termos dados suficientes para simular os *books*. Pelo mesmo motivo, em nossa simulação, as ofertas feitas por estratégias não provocam nenhum impacto no mercado, ou seja, as estratégias não contribuem para a queda ou para o aumento dos preços dos ativos. Acreditamos que nossa abordagem seja uma aproximação razoável dos mercados reais, principalmente quando simulamos seu funcionamento a partir de ativos com grande movimentação, como os da Petrobras, Vale e Dólar.

Para obter dados consistentes, pré-processamos os arquivos visando retirar valores espúrios. Durante a recepção dos dados, alguns problemas ocorreram, fazendo com que valores inviáveis fossem inseridos no histórico. Por exemplo, em alguns instantes, encontramos valores do *bid* negativos ou maiores que o *ask*. Dados com as seguintes propriedades foram excluídos.

– Hora antes das 10h e depois das 17h (horário de funcionamento da BOVESPA & BMF);

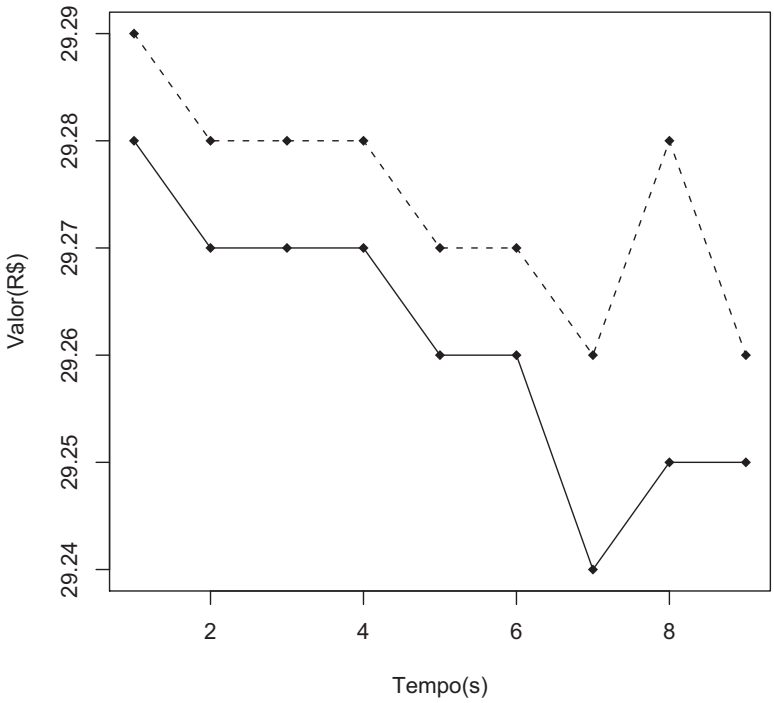

Figura 4.1: Exemplo dos Dados Colhidos

- $bid \geq ask$;
- *Bid*, *ask*, *last*, volume de compra ou volume de venda menor ou igual a zero.

Além disto, em alguns momentos, ocorrem longas repetições de todos os valores das linhas (*bid*, *ask*, *last*, volume de compra e volume de venda). Essas repetições normalmente não ocorrem nos mercados e nos indicam que houve falha na recepção dos dados durante todo o período em que as linhas se repetiram. Retiramos dos dados esses trechos de longas repetições (consideramos longas as repetições que ocorrem por mais de vinte segundos), visto que são provavelmente provenientes de falha.

Os dados foram colhidos segundo a segundo, mas, por alguma falha, não temos os valores em todos os instantes. Encontramos algumas lacunas nos nossos dados. Para facilitar a simulação, preenchemos essas lacunas com os últimos valores válidos. Por exemplo, suponha que temos todos os valores nos instantes 1 e 3. Por algum motivo não temos os valores do instante 2. Para preencher essa lacuna nos dados, repetimos os valores de 1 no instante 2. Se a lacuna for de mais de vinte segundos, não a preenchemos. Preencher essas lacunas ajuda nosso sistema a simular o mercado, já que teremos os valores a

cada segundo e, sendo assim, não corremos o risco de não ter os valores dos ativos em um instante em que alguma estratégia for fazer uma oferta.

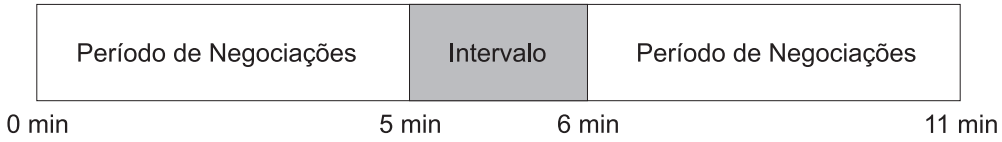

Figura 4.2: Exemplo de Demarcação de Intervalos

Para finalizar o pré-processamento, dividimos cada histórico (Petrobras, Vale e Dólar) em intervalos do mesmo tamanho do horizonte de negociação (H) que utilizamos nos experimentos (cinco minutos). Depois do fim da demarcação de um intervalo, esperamos um minuto para iniciar a demarcação de um novo (como mostrado na Figura 4.2). Os intervalos são sempre de cinco minutos contíguos na linha do tempo. Se, durante a demarcação de um intervalo, for encontrada alguma lacuna, o ponto inicial marcado anteriormente é removido e um novo intervalo se inicia após o final da lacuna.

Seguindo a estratégia de demarcação de intervalos descrita no parágrafo anterior, conseguimos coletar 1350 intervalos a partir dos dados da Petrobras, 938 a partir dos da Vale e 635 a partir dos do Dólar. Esses dados são usados para treinamento e teste das estratégias apresentadas no capítulo 3.

4.2 Medição de Performance

Para analisar o desempenho das estratégias, adotamos uma métrica proposta por Nevmyvaka (Nevmyvaka et al. 2006), que nos permite comparar a performance de estratégias mesmo quando elas negociam ativos diferentes. Como o descrito no capítulo 3, todas as estratégias podem executar apenas uma oferta em cada período de negociação. O desempenho de uma estratégia em uma negociação é o preço da oferta executada dividido pelo *mid spread price* ($(bid + ask)/2$) do primeiro instante do período. Formalizando, se uma estratégia ϵ negocia um ativo em um determinado período p_i e executa uma oferta de valor v, então o desempenho, $\gamma(\epsilon, p_i)$, da estratégia ϵ no período p_i, é dado por:

$$\gamma(\epsilon, p_i) = \frac{v}{\left[bid(p_i) + ask(p_i)\right]/2} \tag{4-1}$$

onde $bid(p_i)$ e $ask(p_i)$ representam o bid e o ask no início de p_i, respectivamente.

Se uma estratégia ϵ for executada em todos os elementos de um conjunto de períodos de negociação $P = \{p_1, p_2, \cdots, p_n\}$, a métrica que reflete o desempenho global da estratégia (denotada por Γ), proveniente de todas as

execuções, pode ser calculada pela média aritmética de cada um dos γ obtidos nos p_i's.

Para execuções de estratégias que objetivam comprar, quanto menor o valor de Γ, mais atrativa a execução é. Isto se deve ao fato de que quanto menor Γ for, menos dinheiro foi pago na compra do ativo. Já no caso da venda, onde o objetivo é acumular capital, quanto maior o valor de Γ, melhor o desempenho da estratégia.

4.3 Estratégias e Parâmetros

Nos experimentos, executamos todas as estratégias descritas no capítulo 3, porém, algumas delas foram executadas com parâmetros diferentes. A estratégia *Reinforcement Learning* (seção 3.4) foi executada de três maneiras diferentes. Na primeira, a executamos apenas com os atributos privados (t,e e f), na segunda, com os privados mais o atributo de estado das filas (q) e na terceira, com os atributos privados mais o atributo público que representa o último contexto que ocorreu (c). Ao usar o atributo c, sempre analisamos um contexto de tamanho dois, ou seja, avaliamos as duas últimas flutuações do valor do ativo para a construção do valor do atributo. Essas execuções nos permitem averiguar o impacto de cada atributo na política gerada pelo algoritmo de aprendizado.

Outra estratégia modificada foi a *Submit and Leave* (seção 3.3). Introduzimos dois parâmetros: $fator$ e $incremento$. Esses parâmetros especificam o valor da oferta inicial da estratégia. Se for uma oferta de compra, o valor inicial será $bid + fator \cdot incremento$ e se for de venda será $ask - fator \cdot incremento$. Em ambos os casos, quanto maior for o $fator$, mais bem posicionada a oferta estará no *book*. Nos ativos da Petrobras e Vale, que usamos para os experimentos, ao parâmetro $incremento$ foi atribuído o valor 0.01, fazendo com que a oferta varie em relação ao bid/ask em apenas um centavo e no histórico dos valores do Dólar, o parâmetro $incremento$ assume o valor 0.5. Durante os experimentos, o parâmetro $fator$ assume todos os valores do conjunto $F = \{-20, -19, \cdots, 0, \cdots, 19, 20\}$, fazendo com que sejam executadas quarenta e uma estratégias *Submit and Leave* em cada experimento.

O único parâmetro configurável da estratégia *Kearns* é a *agressividade*. Executamos a estratégia com cinco valores diferentes para a agressividade: 0, 0.25, 0.5, 0.75 e 1. Uma vez escolhido o valor da agressividade, em todos os pontos da negociação onde a estratégia pode ofertar, a oferta que corresponde à agressividade anteriormente escolhida é selecionada. Por exemplo, se durante uma negociação, a agressividade 1 for selecionada, em todos os R pontos

de tomada de decisão, a oferta que representa a agressividade de valor 1 é escolhida. Durante os experimentos, notamos que, durante um período de negociações, nem sempre a melhor escolha é selecionar ofertas de acordo com um mesmo valor de agressividade. Portanto, implementamos um algoritmo para a seleção dos valores da agressividade a cada momento de decisão. De acordo com a seção 3.5.2, a análise feita pela metodologia estendida gera uma fronteira eficiente de preços para cada um dos R momentos de decisão da estratégia. O algoritmo que implementamos para otimizar a seleção da agressividade a cada momento, seleciona o valor da agressividade que gera a oferta que maximiza a relação $retorno/risco$. O pseudocódigo do algoritmo implementado pode ser visto a seguir.

Algoritmo 5 Otimização de Agressividade

```
1:  A = {0, 0.05, 0.10, 0.15, ··· , 1}
2:  for all f ∈ F do
3:     agressividadeOtima ← 0
4:     maiorQuociente ← −∞
5:     for all ag ∈ A do
6:        razao ← retorno(ag)/risco(ag)
7:        if razao > maiorQuociente then
8:           maiorQuociente ← razao
9:           agressividadeOtima ← ag
10:       end if
11:    end for
12:    AGRS ← agressividadeOtima
13: end for
```

O algoritmo inicia sua execução iterando sobre todas as fronteiras do conjunto F, gerado pela execução de metodologia estendida (linha 1). Para cada fronteira $f \in F$, o algoritmo seleciona a agressividade ag que representa a oferta que maximiza o valor da variável $razao$ (linha 5). As funções $retorno$ e $risco$ retornam os valores do retorno e do risco esperado, respectivamente, para uma determinada agressividade ag. No fim da execução do laço interno (linha 11), o valor da agressividade que otimiza a função objetivo é armazenado na matriz $AGRS$. Durante a execução da estratégia *Kearns* com seleção otimizada de agressividade, a cada momento de decisão, a matriz $AGRS$ é consultada para saber qual agressividade escolher.

Para a execução da estratégia *Aleatória* (seção 3.6), precisamos definir a probabilidade de execução de cada ação da Tabela 3.3. Definimos a probabilidade de se fazer uma oferta igual a de não se fazer, ou seja, $Pr_o = 1/2$. A probabilidade de sorteio de todas as ações que provocam uma oferta também foram definidas igualmente. Do mesmo modo, a probabilidade de execução das

ações que não provocam ofertas foram definidas de maneira igualitária, ou seja, $Pr_3 = Pr_4 = 1/2$.

4.4 Resultados dos Experimentos

Nesta seção, apresentamos os resultados obtidos a partir da execução dos experimentos. Executamos os experimentos usando validação cruzada e o método *holdout*. Fizemos uma validação do tipo *10-fold cross validation* ($K = 10$) no histórico do ativo da Petrobras, outra no da Vale e outra nos dados do Dólar (meses de maio, junho e julho). Além disto, executamos uma validação usando o método *holdout* nos valores históricos do Dólar, usando os meses de maio, junho e julho para treinar e o mês de agosto para testar.

Para a execução de cada *K-fold cross validation*, ordenamos cronologicamente os períodos de negociação. Assim, todos os arquivos de cada *fold*, durante a validação, estão ordenados cronologicamente. Estando os dados usados para validação dispostos dessa maneira, com exceção do ultimo *fold*, as estratégias sempre aprendem, em algum momento, com dados do futuro e testam em dados com data anterior. Esta é uma limitação da *K-fold cross validation* que ocorre quando é aplicada em dados onde existe uma ordem temporal. Nos mercados reais não podemos aprender com dados do futuro para aplicar no passado, mas a aproximação fornecida por esse método de validação parece ser razoável.

Em cada experimento, executamos duas vezes cada estratégia no conjunto de períodos de teste. Na primeira execução, instruímos as estratégias a vender e na segunda, a comprar. Para verificar o impacto da variável R (resolução), que define o número de vezes que as estratégias podem submeter uma oferta ao mercado, executamos os experimentos para três resoluções diferentes: 10, 30 e 60, que permitem que as estratégias façam ofertas a cada 30, 10 e 5 segundos, respectivamente. Na Tabela 4.1, podemos ver a lista dos experimentos que foram executados. Na primeira coluna, vemos o nome do ativo usado no experimento, na segunda vemos o tipo de validação usado, na terceira o objetivo das estratégias no experimento e na última podemos ver a referência para a tabela, localizada no Apêndice A, que contém os resultados detalhados do experimento.

4.5 Observações Gerais sobre os Experimentos

Nesta seção, apresentamos uma análise geral dos resultados obtidos pelas estratégias. Para comparar os resultados obtidos em experimentos diferentes, atribuímos pontos para as estratégias de acordo com suas posições nas tabe-

Ativo	**Tipo de Validação**	R	**Objetivo**	**Resultados**
Dólar	Método *Holdout*	10	Compra	A.4
Dólar	Método *Holdout*	30	Compra	A.5
Dólar	Método *Holdout*	60	Compra	A.6
Dólar	Método *Holdout*	10	Venda	A.7
Dólar	Método *Holdout*	30	Venda	A.8
Dólar	Método *Holdout*	60	Venda	A.9
Dólar	*10-fold cross validation*	10	Compra	A.1
Dólar	*10-fold cross validation*	30	Compra	A.2
Dólar	*10-fold cross validation*	60	Compra	A.3
Dólar	*10-fold cross validation*	10	Venda	A.10
Dólar	*10-fold cross validation*	30	Venda	A.11
Dólar	*10-fold cross validation*	60	Venda	A.12
Vale	*10-fold cross validation*	10	Compra	A.19
Vale	*10-fold cross validation*	30	Compra	A.20
Vale	*10-fold cross validation*	60	Compra	A.21
Vale	*10-fold cross validation*	10	Venda	A.22
Vale	*10-fold cross validation*	30	Venda	A.23
Vale	*10-fold cross validation*	60	Venda	A.24
Petrobras	*10-fold cross validation*	10	Compra	A.13
Petrobras	*10-fold cross validation*	30	Compra	A.14
Petrobras	*10-fold cross validation*	60	Compra	A.15
Petrobras	*10-fold cross validation*	10	Venda	A.16
Petrobras	*10-fold cross validation*	30	Venda	A.17
Petrobras	*10-fold cross validation*	60	Venda	A.18

Tabela 4.1: Lista de Experimentos Executados

las. De acordo com o apresentado na seção 4.3, executamos quarenta e uma instâncias da estratégia *Submit and Leave*, seis da *Kearns* e três da *Reinforcement Learning*. Além disto, executamos uma instância da estratégia Aleatória, outra da Ótima e outra da *Submit and Execute*, totalizando cinquenta e três instâncias de estratégias em cada experimento. O sistema de pontuação funciona da seguinte maneira: a estratégia que conseguiu o primeiro lugar em um experimento recebe cinquenta e três pontos, a que ficou em segundo, cinquenta e dois, a pontuação vai decrescendo até que a estratégia que assumiu o último lugar ganha apenas um ponto. Tal sistema de pontuação não se baseia em nenhuma metodologia conhecida para comparação, ou seja, desenvolvemos esse sistema por julgarmos que seja razoável comparar o desempenho das estratégias entre os experimentos dessa forma. Na Tabela 4.2 podemos ver a classificação das estratégias após a execução de todos os experimentos listados na tabela 4.1. Na Tabela 4.3, são apresentadas as pontuações das estratégias quando apenas os experimentos, com o objetivo de venda, são levados em consideração. A Tabela 4.4 apresenta as somas dos pontos em cada experimento

com o objetivo de compra.

A estratégia que obteve a melhor colocação, desconsiderando a estratégia Ótima, foi a *Kearns*. Em dezessete dos vinte e quatro experimentos, a melhor instância da estratégia *Kearns* ($agressividade = 1$) ficou a frente da *Submit and Execute*. Supondo que os resultados são aleatórios e levando em consideração apenas essas duas estratégias, a probabilidade de conseguirmos pelo menos dezessete vitórias de uma mesma estratégia seria a mesma de se obter *cara* pelo menos dezessete vezes em vinte quatro lançamentos de uma moeda. A probabilidade pode ser calculada da seguinte maneira:

$$P(17) = \frac{\sum_{i=17}^{24} C_{24,i}}{2^{24}} \quad (4\text{-}2)$$

$$= \frac{536155}{16777216} \quad (4\text{-}3)$$

$$= 0,031957328 \quad (4\text{-}4)$$

$$P(17) = 3,19\% \quad (4\text{-}5)$$

A Equação 4-2 divide a soma de todas as possibilidade de obtermos no mínimo dezessete *caras* pelo número total de combinações possíveis de resultados. Efetuando os cálculos podemos ver que, se nossos experimentos fossem eventos aleatórios, a probabilidade de se obter pelo menos dezessete vitórias da estratégia *Kearns* em relação a estratégia *Submit and Execute* é de pouco mais de 3%. Isso posto, é natural afirmar que nossos experimentos não são aleatórios e que existe um real aprendizado das estratégias *Kearns*.

As estratégias *Kearns* reagiram como o esperado, isto é, as estratégias que assumiram maiores riscos ficaram na frente das estratégias que assumiram riscos menores. A estratégia *Kearns (agressividade=1.0)* ficou na segunda posição, já a *Kearns (agressividade=0.75)* ficou na posição vinte e oito, a *Kearns (agressividade=0.5)*, na trigésima quinta, seguida da a *Kearns (agressividade=0.25)* e, por último, a estratégia a *Kearns (agressividade=0.0)* ficou na posição de número quarenta e dois. A estratégia *Kearns* que usa o algoritmo apresentado na seção 4.3 para selecionar as estratégias, ficou classificada na quarta colocação no quadro geral. Analisando os resultados, podemos ver que em aproximadamente 91% dos experimentos (22 de 24) a instância mais agressiva da estratégia *Kearns* ficou a frente das instâncias menos agressivas. Esse desempenho tão superior das intâncias com alta agressividade nos faz perceber que no contexto da estratégia *Kearns*, ser agressivo é mais vantajoso, produzindo mais lucros e, portanto, melhores resultados para tais instâncias.

Em vários experimentos pelo menos uma instância da estratégia *Submit*

and Leave ocupou a segunda colocação, indicando que essa simples estratégia pode ser bem eficiente, porém, para isto, necessita de um ajuste manual do parâmetro *fator*. Esse ajuste deve ser feito cuidadosamente, já que pode levar a estratégia da segunda posição para a última, como na Tabela A.1. A estratégia *Kearns* se baseia exatamente em aprender, pelos históricos, qual seria o melhor *fator* da estratégia *Submit and Leave*. Como detalhado na seção 3.5, a estratégia *Kearns* executa a estratégia *Submit and Leave* em cada um dos R pontos de tomada de decisão. Apesar das estratégias de negociação serem parecidas na *Submit and Leave* e na *Kearns*, o processo de aprendizagem do *fator* da estratégia *Kearns* faz com que ela seja mais poderosa e eficiente, como podemos ver no quadro de pontuação mostrado na Tabela 4.2.

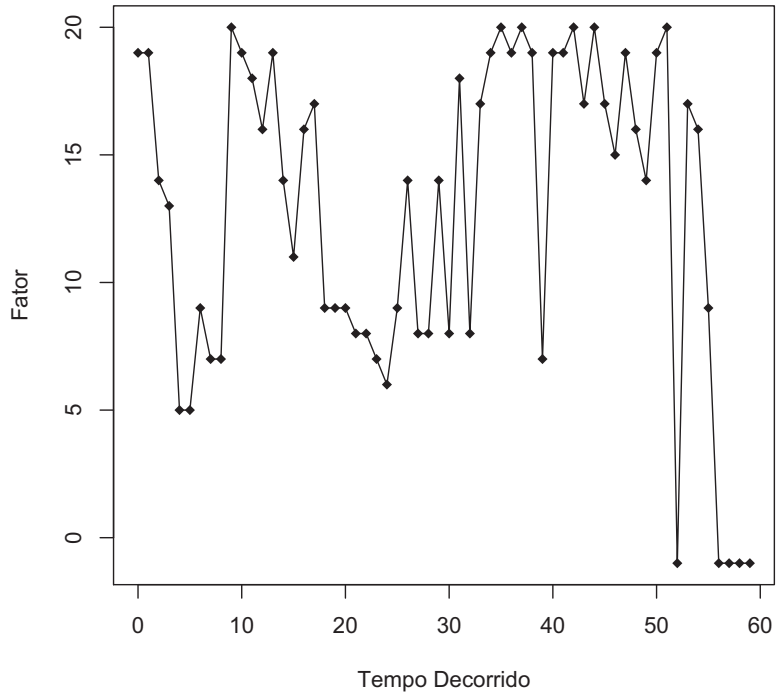

Figura 4.3: Política Kearns ($agressividade = 1.0$, $R = 60$ e objetivo de compra)

O processo de aprendizado da estratégia *Kearns* gera uma tabela de fatores de oferta para cada ponto de decisão. Na Figura 4.3, podemos ver o fator aprendido, a partir dos dados da Petrobras, em cada ponto de decisão para negociações de compra com resolução 60. Pelo gráfico, podemos ver que o parâmetro *fator* segue uma tendência de decréscimo, ou seja, durante a compra, quanto mais perto do fim do período de negociações, maior o preço da oferta. É natural que esse tipo de comportamento aconteça, já que se a

estratégia não consegue comprar por preços mais baixos ao passar do tempo, aumentar o preço de oferta também aumenta a probabilidade de execução da oferta.

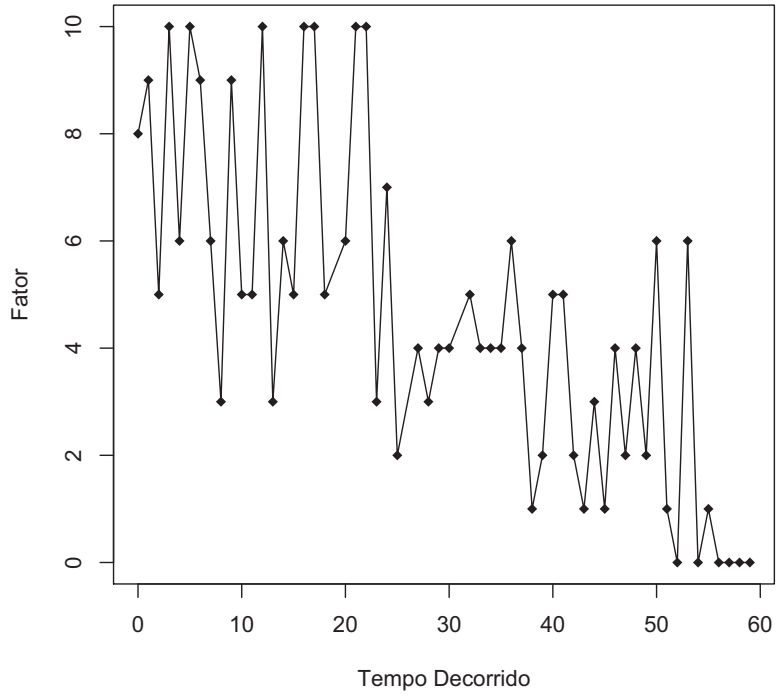

Figura 4.4: Política *Reinforcement Learning* ($R = 60$, objetivo de compra)

À estratégia *Reinforcement Learning* é permitido, nos momentos de decisão, não fazer ofertas. Isto ocorre se a ação A_3 ou A_4 da Tabela 3.3 for escolhida. Em todos os outros momentos, uma oferta é feita. Da mesma maneira que a estratégia *Kearns*, um fator de incremento em relação ao *bid* (se for uma negociação de compra) é escolhido. Na Figura 4.4, podemos ver um gráfico com todos os fatores escolhidos na fase de aprendizado da estratégia *Reinforcement Learning*. O algoritmo de aprendizado, nesse exemplo, foi executado usando os dados da Petrobras para uma negociação de compra, com $R = 60$ e os atributos utilizados na fase de aprendizado foram t, f e e. Plotamos no gráfico apenas os fatores nos pontos onde fazer uma oferta foi escolhido. Mais claramente, podemos ver que a mesma tendência de decréscimo apresentada anteriormente é seguida, confirmando o fato de que, ao passar do tempo, aumentar o preço da oferta é vantajoso. Tanto na estratégia *Kearns* como na *Reinforcement Learning*, esse mesmo tipo de comportamento foi observado quando o objetivo é vender.

As várias instâncias da estratégia *Submit and Leave* recebem retornos

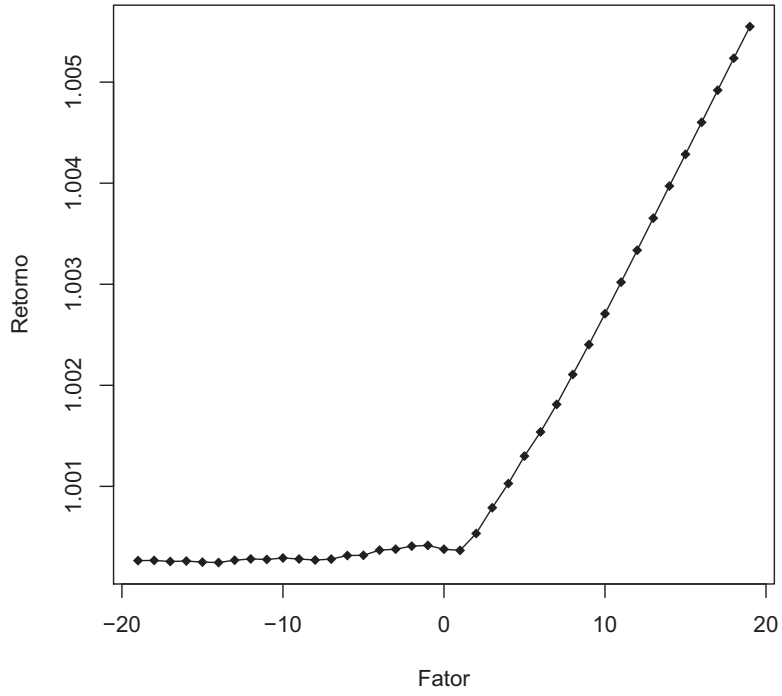

Figura 4.5: Retornos recebidos em função do fator (Compra)

bem diferentes em função do parâmetro *fator* escolhido. Na Figura 4.5, podemos ver a média aritmética dos retornos obtidos por cada instância nos experimentos de compra e na Figrura 4.6, nos de venda. Em ambas as curvas, os retornos médios vão se tornando menos atrativos (altos na compra e baixos na venda) com o aumento do fator. Além disto, o retorno médio máximo recebido é dado por instâncias diferentes nos dois gráficos, mostrando a importância de se escolher corretamente o valor do parâmetro *fator*. Pelos gráficos, podemos concluir que a estratégia *Submit And Leave* pode obter retornos atrativos, mas precisa do ajuste correto do fator, que não é algo trivial, já que uma pequena mudança no fator pode acarretar em perdas significativas. Analisando o gráfico, podemos ver que fatores acima de zero não são atrativos, o que nos diz que, no momento da escolha do fator, escolher fatores menores que zero aumenta a probabilidade de bons retornos.

Executamos três instâncias da estratégia *Reinforcement Learning*, uma usando apenas os atributos privados (e, f e t), outra usando os privados mais o atributo de contexto (e, f, t e c) e outra usando os privados junto com o atributo de estado das filas (e, f, t e q). A instância que obteve melhor desempenho foi a que usou apenas os atributos privados, ocupando a vigésima posição. Em seguida, na vigésima segunda colocação, figura a instância que usa

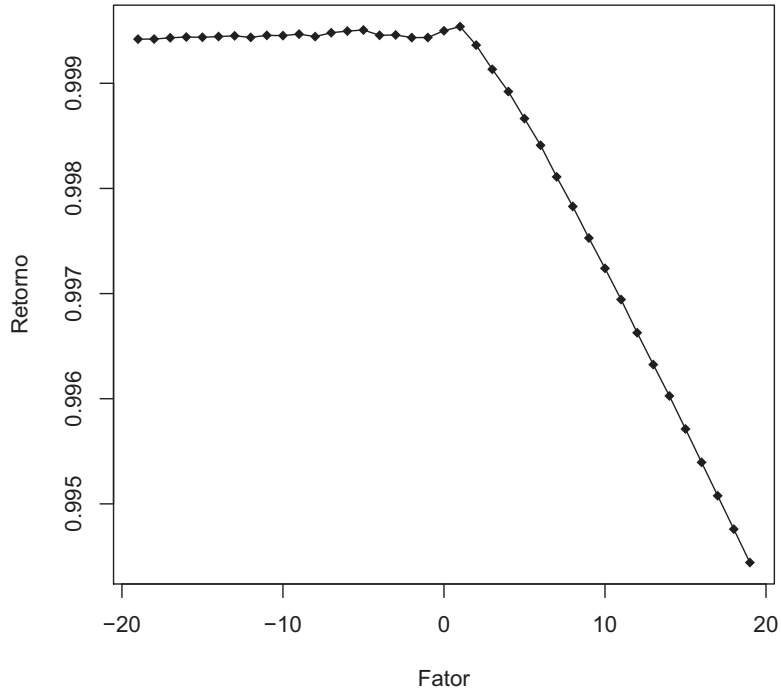

Figura 4.6: Retornos recebidos em função do fator (Venda)

o atributo c, que vem uma posição acima da instância que usa o estado das filas para aprendizado. As três instâncias obtiveram resultados semelhantes, porém, a que possui menos estados, ficou na frente. Dos dois atributos públicos que inserimos, o que mais ajudou a estratégia foi o contexto (c). Apesar de ser o atributo que mais aumenta o número de estados do ambiente, dificultando o aprendizado, ainda assim a instância que o usa conseguiu ficar na frente da estratégia que faz uso de q.

Quando usamos apenas os atributos e, f e t, todos os estados são válidos em todos os períodos de negociação usados para treino. Assim, essa instância usa 100% dos dados disponiveis para treino. A instância que usa o atributo c só consegue aprender em média em 10% dos intervalos separados para treino, já que nem todos os estados são válidos em todos os períodos (Algoritmo 3), e a que usa q, consegue aprender em 50% dos casos. A instância melhor posicionada obteve um bom desempenho por usar todos os arquivos separados para treino. Se existissem mais dados para treino, é de se esperar que as outras instâncias melhorariam seus desempenhos, já que poderiam treinar mais vezes cada estado.

A estratégia Aleatória executa as mesmas ações que as executadas pelas instâncias da estratégia *Reinforcement Learning*, só que de maneira randômica.

Basicamente, a estratégia Aleatória é uma instância de *Reinforcement Learning* que não aprendeu nenhuma política de negociação e, por isso, escolhe ao acaso, em cada momento de oferta, uma ação a ser executada. Essa estratégia foi introduzida para que pudéssemos analisar a eficácia do processo de aprendizagem das estratégias *Reinforcement Learning*. Observando os resultados, pudemos constatar que em 22 dos 24 experimentos ($\approx 91{,}6\%$) todas as instâncias da estratégia *Reinforcement Learning* obtiveram melhor desempenho que a estratégia Aleatória. Utilizando o mesmo raciocínio para cálculo de aleatoriedade de eventos apresentado anteriormente, podemos chegar a conclusão de que a probabilidade de uma estratégia ficar a frente de outra em vinte e duas de vinte e quatro vezes possíveis é de $0,0017\%$ (equação 4-6). A partir desses dados, podemos concluir que as estratégias *Reinforcement Learning* conseguem, no processo de aprendizado, absorver informações importantes do mercado, ou seja, existe um real aprendizado por parte dessas estratégias. Portanto, a sequência de ações aprendidas são mais eficazes que a submissão aleatória de ações ao ambiente.

$$P(22) = \frac{\sum_{i=22}^{24} C_{24,i}}{2^{24}} \tag{4-6}$$

$$= \frac{301}{16777216} \tag{4-7}$$

$$= 0,00001794 \tag{4-8}$$

$$P(22) = 0,0017\% \tag{4-9}$$

De acordo com o que foi descrito anteriormente, executamos experimentos nos mesmos conjuntos de dados alterando o valor da variável *resolução*. Com a alteração desse valor, podemos diminuir ou aumentar o número máximo de ofertas que uma estratégia pode fazer durante um determinado período de negociações. Em alguns experimentos, aumentando a resolução conseguimos melhorar o desempenho de estratégias como a *Reinforcement Learning*, já em outros, o aumento da resolução foi prejudicial para instâncias dessa mesma e de outras estratégias. Não conseguimos detectar nenhum padrão na variação dos desempenhos das estratégias no que diz respeito a variação da resolução. Portanto, chegamos a conclusão que nossos experimentos são inconclusivos em relação a alteração do valor dessa variável.

Posição	**Estratégia**	**Pontos**
1	Ótima	1272.0
2	Kearns (agressividade =1.0)	1044.0
3	Submit and Leave (fator=-14.0)	1020.0
4	Kearns (agressividade otimizada)	1017.0
5	Submit and Leave (fator=-15.0)	1000.0
6	Submit and Leave (fator=-16.0)	990.0
7	Submit and Leave (fator=-13.0)	990.0
8	Submit and Leave (fator=-17.0)	973.0
9	Submit and Leave (fator=-18.0)	970.0
10	Submit and Leave (fator=-19.0)	962.0
11	Submit and Leave (fator=-9.0)	954.0
12	Submit and Leave (fator=-20.0)	952.0
13	Submit and Leave (fator=-11.0)	944.0
14	Submit and Leave (fator=-7.0)	936.0
15	Submit and Leave (fator=-8.0)	928.0
16	Submit and Leave (fator=-6.0)	924.0
17	Submit and Leave (fator=-5.0)	918.0
18	Submit and Leave (fator=1.0)	904.0
19	Submit and Leave (fator=-12.0)	900.0
20	Reinforcement Learning (e,f,t)	896.0
21	Submit and Leave (fator=-10.0)	879.0
22	Reinforcement Learning (c,e,f,t)	869.0
23	Reinforcement Learning (e,f,q,t)	860.0
24	Submit and Execute	819.0
25	Submit and Leave (fator=-4.0)	802.0
26	Submit and Leave (fator=-3.0)	755.0
27	Submit and Leave (fator=0.0)	739.0
28	Kearns (agressividade =0.75)	680.0
29	Submit and Leave (fator=-2.0)	678.0
30	Submit and Leave (fator=-1.0)	663.0
31	Submit and Leave (fator=2.0)	655.0
32	Aleatória	570.0
33	Submit and Leave (fator=3.0)	556.0

Continua na próxima página

Posição	Estratégia	Pontos
34	Submit and Leave (fator=4.0)	522.0
35	Kearns (agressividade =0.5)	497.0
36	Kearns (agressividade =0.25)	471.0
37	Submit and Leave (fator=5.0)	464.0
38	Submit and Leave (fator=6.0)	412.0
39	Submit and Leave (fator=7.0)	364.0
40	Submit and Leave (fator=8.0)	333.0
41	Submit and Leave (fator=9.0)	309.0
42	Kearns (agressividade =0.0)	299.0
43	Submit and Leave (fator=10.0)	280.0
44	Submit and Leave (fator=11.0)	255.0
45	Submit and Leave (fator=12.0)	231.0
46	Submit and Leave (fator=13.0)	204.0
47	Submit and Leave (fator=14.0)	177.0
48	Submit and Leave (fator=15.0)	147.0
49	Submit and Leave (fator=16.0)	120.0
50	Submit and Leave (fator=17.0)	96.0
51	Submit and Leave (fator=18.0)	72.0
52	Submit and Leave (fator=19.0)	48.0
53	Submit and Leave (fator=20.0)	24.0

Tabela 4.2: Pontuação Geral das Estratégias

Posição	Estratégia	Pontos
1	Ótima	636.0
2	Submit and Leave (fator=-9.0)	501.0
3	Submit and Leave (fator=-13.0)	496.0
4	Submit and Leave (fator=-11.0)	490.0
5	Submit and Leave (fator=-14.0)	487.0
6	Submit and Leave (fator=-16.0)	484.0
7	Submit and Leave (fator=1.0)	484.0
8	Submit and Leave (fator=-5.0)	483.0
9	Submit and Leave (fator=-6.0)	482.0
10	Kearns (agressividade =1.0)	476.0
11	Submit and Leave (fator=-15.0)	474.0
12	Reinforcement Learning (e,f,t)	462.0
13	Kearns (agressividade otimizada)	462.0
14	Submit and Leave (fator=-7.0)	460.0
Continua na próxima página		

Posição	**Estratégia**	**Pontos**
15	Submit and Leave (fator=-10.0)	454.0
16	Submit and Leave (fator=-17.0)	452.0
17	Submit and Execute	448.0
18	Submit and Leave (fator=-12.0)	445.0
19	Submit and Leave (fator=-20.0)	436.0
20	Submit and Leave (fator=-18.0)	436.0
21	Reinforcement Learning (e,f,q,t)	433.0
22	Submit and Leave (fator=-19.0)	427.0
23	Submit and Leave (fator=-8.0)	425.0
24	Reinforcement Learning (c,e,f,t)	412.0
25	Submit and Leave (fator=-3.0)	408.0
26	Submit and Leave (fator=0.0)	408.0
27	Submit and Leave (fator=-4.0)	396.0
28	Submit and Leave (fator=2.0)	370.0
29	Submit and Leave (fator=-2.0)	361.0
30	Submit and Leave (fator=-1.0)	358.0
31	Kearns (agressividade =0.75)	336.0
32	Submit and Leave (fator=3.0)	323.0
33	Aleatória	316.0
34	Submit and Leave (fator=4.0)	303.0
35	Submit and Leave (fator=5.0)	262.0
36	Submit and Leave (fator=6.0)	223.0
37	Kearns (agressividade =0.5)	213.0
38	Submit and Leave (fator=7.0)	190.0
39	Kearns (agressividade =0.25)	175.0
40	Submit and Leave (fator=8.0)	171.0
41	Submit and Leave (fator=9.0)	159.0
42	Submit and Leave (fator=10.0)	145.0
43	Submit and Leave (fator=11.0)	132.0
44	Submit and Leave (fator=12.0)	120.0
45	Kearns (agressividade =0.0)	108.0
46	Submit and Leave (fator=13.0)	105.0
47	Submit and Leave (fator=14.0)	90.0
48	Submit and Leave (fator=15.0)	75.0
49	Submit and Leave (fator=16.0)	60.0
50	Submit and Leave (fator=17.0)	48.0

Continua na próxima página

Posição	Estratégia	Pontos
51	Submit and Leave (fator=18.0)	36.0
52	Submit and Leave (fator=19.0)	24.0
53	Submit and Leave (fator=20.0)	12.0

Tabela 4.3: Pontuação das Estratégias (Venda)

Posição	Estratégia	Pontos
1	Ótima	636.0
2	Kearns (agressividade =1.0)	568.0
3	Kearns (agressividade otimizada)	555.0
4	Submit and Leave (fator=-19.0)	535.0
5	Submit and Leave (fator=-18.0)	534.0
6	Submit and Leave (fator=-14.0)	533.0
7	Submit and Leave (fator=-15.0)	526.0
8	Submit and Leave (fator=-17.0)	521.0
9	Submit and Leave (fator=-20.0)	516.0
10	Submit and Leave (fator=-16.0)	506.0
11	Submit and Leave (fator=-8.0)	503.0
12	Submit and Leave (fator=-13.0)	494.0
13	Submit and Leave (fator=-7.0)	476.0
14	Reinforcement Learning (c,e,f,t)	457.0
15	Submit and Leave (fator=-12.0)	455.0
16	Submit and Leave (fator=-11.0)	454.0
17	Submit and Leave (fator=-9.0)	453.0
18	Submit and Leave (fator=-6.0)	442.0
19	Submit and Leave (fator=-5.0)	435.0
20	Reinforcement Learning (e,f,t)	434.0
21	Reinforcement Learning (e,f,q,t)	427.0
22	Submit and Leave (fator=-10.0)	425.0
23	Submit and Leave (fator=1.0)	420.0
24	Submit and Leave (fator=-4.0)	406.0
25	Submit and Execute	371.0
26	Submit and Leave (fator=-3.0)	347.0
27	Kearns (agressividade =0.75)	344.0
28	Submit and Leave (fator=0.0)	331.0

Continua na próxima página

Posição	**Estratégia**	**Pontos**
29	Submit and Leave (fator=-2.0)	317.0
30	Submit and Leave (fator=-1.0)	305.0
31	Kearns (agressividade =0.25)	296.0
32	Submit and Leave (fator=2.0)	285.0
33	Kearns (agressividade =0.5)	284.0
34	Aleatória	254.0
35	Submit and Leave (fator=3.0)	233.0
36	Submit and Leave (fator=4.0)	219.0
37	Submit and Leave (fator=5.0)	202.0
38	Kearns (agressividade =0.0)	191.0
39	Submit and Leave (fator=6.0)	189.0
40	Submit and Leave (fator=7.0)	174.0
41	Submit and Leave (fator=8.0)	162.0
42	Submit and Leave (fator=9.0)	150.0
43	Submit and Leave (fator=10.0)	135.0
44	Submit and Leave (fator=11.0)	123.0
45	Submit and Leave (fator=12.0)	111.0
46	Submit and Leave (fator=13.0)	99.0
47	Submit and Leave (fator=14.0)	87.0
48	Submit and Leave (fator=15.0)	72.0
49	Submit and Leave (fator=16.0)	60.0
50	Submit and Leave (fator=17.0)	48.0
51	Submit and Leave (fator=18.0)	36.0
52	Submit and Leave (fator=19.0)	24.0
53	Submit and Leave (fator=20.0)	12.0

Tabela 4.4: Pontuação das Estratégias (Compra)