Soft Comput (2016) 20:5051–5066
DOI 10.1007/s00500-016-2298-4

METHODOLOGIES AND APPLICATION

Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach

Saeid Fallahpour[1] · **Hasan Hakimian**[1] · **Khalil Taheri**[2] · **Ehsan Ramezanifar**[3]

Published online: 8 August 2016

Abstract Recent studies show that the popularity of the pairs trading strategy has been growing and it may pose a problem as the opportunities to trade become much smaller. Therefore, the optimization of pairs trading strategy has gained widespread attention among high-frequency traders. In this paper, using reinforcement learning, we examine the optimum level of pairs trading specifications over time. More specifically, the reinforcement learning agent chooses the optimum level of parameters of pairs trading to maximize the objective function. Results are obtained by applying a combination of the reinforcement learning method and cointegration approach. We find that boosting pairs trading specifications by using the proposed approach significantly overperform the previous methods. Empirical results based on the comprehensive intraday data which are obtained from S&P500 constituent stocks confirm the efficiently of our proposed method.

Communicated by V. Loia.

✉ Hasan Hakimian
hasan.hakimian@ut.ac.ir

Saeid Fallahpour
falahpor@ut.ac.ir

Khalil Taheri
k.taheri@ut.ac.ir

Ehsan Ramezanifar
e.ramezanifar@maastrichtuniversity.nl

[1] Department of Finance, Faculty of Management, University of Tehran, Tehran, Iran

[2] Advanced Robotics and Intelligent Systems Laboratory, School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran

[3] Department of Finance, School of Business and Economics, Maastricht, The Netherlands

Keywords Pairs trading · Reinforcement learning · Cointegration · Sortino ratio · Mean-reverting process

1 Introduction

The idea of pairs trading relies on long-term equilibrium among a pair of stocks which is used by many high-frequency traders. Growing popularity of this strategy may be a reason for losing arbitrage opportunities and lead the traders to have a lower chance to take a handsome profit from this strategy. Hence, optimization of pairs trading strategy has gained widespread attention among researchers and practitioners. Using the reinforcement learning, we optimize specifications of pairs trading through the cointegration approach. In addition, to exemplify the application of this method, we use the comprehensive recent intraday data of US equity market from June 2015 to January 2016. Although pairs trading strategy has been the subject of extensive research, no prior studies employ the reinforcement learning to optimize the performance of pairs trading strategy; the results show that our optimizing strategy is significantly overperformed the prior methods.

Pairs trading strategy is a market neutral strategy. Hence, regardless of market movement, the profit of this strategy ultimately depends on the change in the difference in price of two stocks. This strategy employs the statistical arbitrage opportunities raised from the temporary fluctuations in the price of two assets which in long-term are in equilibrium (Gatev et al. 2006).

When these prices temporarily deviate from the equilibrium, the strategy opens a long position on the asset which is overpriced and a short position on the asset which is underpriced. When the prices have returned to their long-

term equilibrium, the positions are changed and the profit is obtained.

As Gatev et al. (2006) discussed, the growing popularity of the pairs trading strategy may also pose a problem as the opportunities to trade become much smaller.

There are four well-known approaches of pairs trading strategies: distance approach, combine forecast approach, stochastic approach, and cointegration approach.

In this study, we employ the cointegration approach. The first step for implementing a strategy is to select a pair of assets which have long-run statistical relationships. In fact, this statistical relationship during the trading period is the necessary condition for conducting transactions on a pair of assets. Since new data are always put into the trading system during the transactions, the desired statistical relationships should be checked through taking appropriate estimation and executive time windows.

After selecting the pair of stocks through examining the existence of cointegration relationship, the model parameters are estimated. Finally, the model is allocated to execute the trade, and the standardized value of the spread resulting from the discrepancy between prices of the assets is plotted to conduct transactions and select the appropriate positions. The value of spread fluctuates on both sides of the mean value due to the mean reversion feature and the long-run equilibrium relationships. The spread thresholds at which the trading position should be opened and the amount of stop-loss are specified as two boundaries on either side of the mean spread value. The distance of boundaries from the mean spread specifies the profit of each trade and shows that how long each trade should continue which means from opening the position to closing it. Overall, selecting the appropriate pairs trading specifications such as executive time windows and boundaries plays an important role in the profitability of this strategy. After constructing the portfolios of two stocks' candidate, the Sortino ratio employs as an objective function to evaluate the performance of the portfolios. Therefore, we attempt to maximize the objective function by applying a series of nonlinear limitations for obtaining an optimal model.

We next combine the cointegration approach with the reinforcement learning to optimize its strategy features. The main idea behind the reinforcement learning is actually a penalty–reward strategy. Reinforcement learning consists of two main components, namely "agent" and "environment." The agent lives in the environment and updates its experiences based on the feedback it receives from the environment. The reinforcement learning is used in our study, to optimally determine the parameter estimation window, executive time windows, trading thresholds, and the stop-loss boundary to maximize the Sortino ratio. In other words, the strategy of pairs trading is modeled with one kind of reinforcement learning problems which is named N-arm bandit. N-arm bandit is basically a game device consisting of N-arm and by pulling a lever/arm, the player is rewarded with a score from the device. Each player can pull only a limited number of arms and receive a score he/she has gained in each pulling. The goal of the player is in maximizing the summation of the acquired scores from the game device in the limited iterations, i.e., pulling lever/arm. This problem is used in reinforcement learning for designing a knowledge-acquisitive agent. We show that, similar pulling an arm in each iteration of N-arm bandit problem, the agent selects values for the four desired parameters, i.e., the arms in N-arm bandit, when a portfolio is opened, i.e., the iteration in our problem, based on its learning from and exploration in the environment, i.e., long-term relationship of two candidate stocks. The goal of the agent is to maximize the Sortino objective function, i.e., the received score from the environment. The results indicated the superiority of this algorithm in simultaneously improving the return and reducing the negative risks as compared with the previous algorithms.

The rest of the paper consists of the following sections. A comprehensive investigation of the relevant studies is presented in Sect. 2. Section 3 explains the data. The methodology including the concepts of cointegration and pairs trading, and the reinforcement learning is mentioned in Sect. 4. Finally, Sects. 5 and 6 describe results/discussion and conclusion, respectively.

2 Relevant literature

Many studies conducted on pairs trading are concentrated on the efficiency and performance of the pairs trading strategy in the foreign markets, for example Gatev et al. (2006). They used the daily US stock return data to test the strategy of pairs trading from 1962 until 2002. Using a simple trading rule, they calculated annual excess returns of over 11 % for each year of the entire sample period. Kawasaki et al. (2003) used the cointegration approach in pairs trading to find the stock pairs in the Tokyo Stock Exchange. Linking the uninformed demand stocks with the interests and risks of pairs trading, Andrade et al. (2005) investigated the profitability of pairs trading in the Taiwan stock exchange. Perlin (2009) studied the performance and risk of pairs trading in the Brazilian financial market under different frequencies of the database during a 1-year period. Muslumov et al. (2009) used the distance approach to test pairs trading in the Istanbul stock exchange and did not consider any limitations such as the type of industry or measurement in the formation process of pairs so that each stock could pair up with another only under the condition of minimum distance. This methodology had an excess return of 5.4 % for 20 portfolios of pairs trading on average. Bogomolov (2011) compared the profitability of pairs trading strategy in Australia stock exchange

through the distance approach, the cointegration method, and the statistical spread method.

Huck and Afawubo (2015), by using the components of the S&P 500 index, explored the performance of a pairs trading system based on various pairs' selection methods. Although large empirical applications in the literature focus on the distance method, their proposed approach also deals with well-known statistical and econometric techniques such as stationary and cointegration which make the trading system much more affordable from a computational point of view. By controlling the risk and transaction costs, the results confirmed that the distance method generates insignificant excess returns. Their approach revealed that cointegration provides a high, stable, and robust return, and an approach leads to a weak performance if a pair selection uses the stationary criterion. We also used cointegration approach in our research to implement pairs trading strategy using the reinforcement learning method.

Designing the trading regulations and high-frequency trading strategies has been studied for many years. Zhang (2001) carried out a research for designing the strategies of high-frequency trading in relation with certain relevant parameters including selection of trading boundaries limits, applying position of stop-loss boundaries, and the calculation method of the return. Zhang (2001) specified a selling rule based on two boundary levels, one target price, and one stop-loss limit. In order to achieve optimal boundary levels, Zhang (2001) solved a set of two-point boundary value problems. Through a model called switching geometric Brownian motion, Guo and Zhang (2005) studied the optimal selling rule. They derived optimal selling rules for an investor holding one stock in a market which fluctuates among several states. The mathematical model for stock fluctuations is a regime switching in which one set of the established Black–Scholes models coupled with a finite-state continuous Markov chain. The optimal stopping rule is a threshold type for each state, derived via the modified smooth fit.

de Moura et al. (2016) proposed a pairs trading strategy entirely based on linear state space models designed for modeling the spread formed with a pair of assets. Once an adequate state space model for the spread is estimated, they used the Kalman filter to calculate conditional probabilities that the spread will equal to its mean long-term value. The strategy is activated upon large values of these conditional probabilities: the spread is bought or sold accordingly. They have evaluated their approach on the data (collected on September 22, 2011, to March 26, 2013) on daily stock prices of two securities, Exxon Mobil Corporation (traded in the NYSE with the symbol XOM) and Southwest Airlines Co (traded in the NYSE with the symbol LUV). The main difference of this work with our research is that the security price uses a state space models, while our method uses a cointegration approach.

By using the stochastic control approach, Tourin and Yan (2013) proposed a model for analyzing dynamic pairs trading strategies. It is assumed that a portfolio consists of a bank account and two co-integrated stocks and the model tries to explore an optimal portfolio setting. The goal of the approach is to maximize the cumulative profit for a fixed time horizon and the expected terminal utility of wealth. Moreover, they have evaluated their approach on the data (collected on October 17, 2011, minute-by-minute), on two stocks traded on the New York Stock Exchange, Goldman Sachs Group, Incorporated, with ticker symbols GS, and J.P. Morgan Chase and Company, with ticker symbol JPM. The results showed that the approach maximizes cumulative profit and minimizes the loss function. The major concern about this work is related to uncertainty of the stochastic control approaches. On the other hand, the approach can find the optimal solution which is remarkable.

Regarding the pairs trading systems, Puspaningrum et al. (2010) studied the pre-set boundaries which were selected in opening the positions in pairs trading. They also studied their impact on the minimum total profit of a specific trading horizon considering the time and an analytical formula. They developed a numerical algorithm to estimate the average trade duration, the average inter-trade interval, and the average number of trades and then use them to find the optimal pre-set boundaries that would maximize the minimum total profit for cointegration error. Moreover, they have examined their algorithm on the seven share pairs (ANZ-ADB, ABC-HAN, ABC-BLD, CCL-CHB, HAN-RIN, BHP-RIO, and TNS-TVL) from the Australian Stock Exchange using daily data for 2004. The numerical result on the stock pair BHP-RIO shows that minimum total profit is $12.96 and 1 trade (either a U-trade or an L-trade) per 15.70 days. Bertram (2010) selected the appropriate boundaries for a combined asset which would follow OU process. Bertram (2010) observed that the optimal boundaries were symmetrically placed around the mean for both return maximization per each time unit and the Sharpe ratio maximization. To illustrate the results, he used combination of the dual-listed securities for ANZ Banking Group Ltd (ANZ.AX, ANZ.NZ). These securities are traded on the Australian and New Zealand stock exchanges simultaneously. Then, the security price in this work follows an Ornstein–Uhlenbeck process. Based on the impact of selecting narrow and wide boundaries on the return and period of each trade, Zeng and Lee (2014) discussed and studied the optimal boundaries with a function of trading costs and parameters of OU process in order to maximize the average profit expected in the long run. By implementing their method on the daily data of Coca Cola and Pepsi, they observed that the new strategy proposed by them had a better performance than the common exercises. Their security price follows an Ornstein–Uhlenbeck process.

With the help of classical mean–variance portfolio selection criterion, Chiu and Wong (2015) introduced the optimal dynamic trading of cointegrated assets. The optimal strategy is acquired over the set of time-consistent policies to ensure rational economic decisions. From a nonlinear Hamilton–Jacobi–Bellman partial differential equation, they solved the optimal dynamic trading strategy in a closed-form explicit solution. They mentioned some examples for evaluating their approach in which cointegrated assets show to be a persistent approach, while by using pre-commitment trading strategies, it would be possible to generate infinite leverage when a cointegrating factor of the assets has a high mean reversion rate. The main drawback of their approach is that it could not get best results in long-term investments. However, their approach can find the best solution in most of the cases.

It should be noted that the above-mentioned methods used fixed parameters values for time windows, trading thresholds, and stop-loss. In this paper, we use a dynamic approach to find the best model parameters including time windows, trading threshold, and also stop-loss to maximize the objective function. For this purpose, we use the reinforcement learning approach. The methods which used the reinforcement learning approaches are described next.

Reinforcement learning is a computational approach which understands, learns, and makes decisions automatically and through environmental inspiration toward its objective. This algorithm has been distinguished from other computational approaches in terms of emphasis on unique learning of direct interactions with the environment. Reinforcement learning was stated as the third category of learning algorithms. In this type of learning, it is actually clarified how different positions are mapped to different actions in order to maximize the obtained profit. The agent designed with the reinforcement learning tries to obtain the expected action which maximizes the profit through experience instead of being told what to do. This agent can consider different policies to select the desired action while making decisions. One example of common policies to select the action is the ϵ-$greedy$ policy which selects the action in ϵ with the highest value in terms of the agent and then selects the actions randomly and independently of their value in $1-\epsilon$ (Sutton and Barto 1998).

Based on the learning power, the reinforcement learning algorithm is able to mathematically formulate a set of variables about which there is no knowledge or predefinition pertaining to their environment and structure. In financial analyses and problems where uncertainty and dynamics are considered to be essential components, this algorithm can be very useful (Sutton and Barto 1998).

By implementing the absolute profit and relative risk-adjusted profit (Sharpe ratio) as the performance function, Gao and Chan (2000) proposed a hybrid of the Q-learning and the Sharpe ratio maximization algorithms for portfolio trades and management to test their model. They indicated the profitability of their algorithm by trading in the external stock markets. The major concern of this approach is that there is no guarantees for Q-learning to converge to a fixed point. In contrast, by experiencing different characteristics of the environment, their approach could converge to a good solution. Lee et al. (2007) proposed a stock trading multi-agent Q-learning framework in order to improve the performance of the systems based on the reinforcement learning. They engaged in stocks trading by defining the necessary roles to make selective decisions and price the stocks simultaneously. Checking the test results obtained by implementing their trading framework on Korea Stock Exchange, they observed a better performance in comparison with other similar approaches. The disadvantages and advantages of this approach are similar to the previous one that has been already mentioned. Won Lee (2001) used the reinforcement learning to model and to learn different types of interactions in real situations for the problem of predicting the stock price. He considered the problem of predicting the stock price as a Markov's process which can be optimized on the basis of the reinforcement learning algorithm. Won Lee (2001) used the stock price trends and consecutive price changes to describe the state in the reinforcement learning. Although the assumption of Markov's process would simplify the problem, it can lead to losing some information. Tan et al. (2011) used adaptive network fuzzy inference system (ANFIS) accompanied by reinforcement learning in order to propose a non-arbitrage high-frequency trading system. They examined the approach on several industries data from 1994 to 2005. The results showed that their approach can perform more efficient than the previous work. It should be noted that ANFIS network can be overfitted, but they have the ability to detect all possible interactions between random variables. Moody and Saffell (2001) proposed a method for optimizing the portfolio and trading systems based on direct reinforcement learning to discover investment policies. Their method was an adaptive algorithm called recurrent reinforcement learning (RRL). They used the Sharpe ratio and negative deviations to optimize the portfolio in their research. They evaluated their proposed approach on an intradaily currency trader and a monthly asset allocation system for the S&P 500 Stock Index and T-Bills. The primitive results showed that the approach can perform efficiently. As they used Q-learning and TD-learning approaches for reinforcement learning, their proposed method is similar to the previous work.

Huang et al. (2015) proposed a new approach for pairs trading using genetic algorithms (GA). They evaluated their approach up to 10 different stock markets. The results showed that the approach significantly outperforms the benchmark. Moreover, their proposed method can generate robust models to cope with the dynamic characteristics in the financial issues. In one hand, the GA can stick in local minima, and

Table 1 Summary statistics of the entire sample

	Mean	SD	Min	Max	Obs.
Facebook	97.12	7.04	74.10	115.68	61,298
Google	664.08	71.84	515.35	779.71	61,862
Amazon	611.58	58.17	474.47	722.25	65,536
Twitter	20.63	5.29	13.77	31.84	65,536
Alibaba	75.56	7.82	57.35	87.71	61,350
General motors	32.38	2.44	25	36.87	61,345
Ford motors	14.17	0.905	11.23	15.83	61,337
Center Point Energy	18.25	0.909	16.07	20.09	63,949
Next Era Energy	101.80	3.63	94.09	113.42	64,194
Dow Chemical	48.12	4.08	36.38	57.08	63,797
Lyondell Basell	90.32	7.89	69.12	106.49	63,774
Adobe Systems Inc	85.32	5.11	71.64	96.4	64,081
Autodesk Inc	53.68	5.80	42.09	65.74	63,811
Amphenol Corp A	53.25	2.87	45.11	58.63	63,933
Boeing Company	139.51	7.52	115.09	150.54	63,885
Johnson & Johnson	98.69	3.34	83.08	105.48	63,687
Medtronic plc	74.65	2.87	61.15	78.90	63,798
Bank of America Corp	16.63	1.16	12.94	18.47	63,777
The Bank of New York Mellon Corp	39.92	0.58	39	41.42	36,685
Helmerich & Payne	56.53	7.36	40.02	75.21	63,892
Hess Corporation	56.6	8.02	32.44	70.6	63,586
The Hershey Company	89.74	3.22	82.46	97.4	63,912
Tyson Foods	46.09	4.23	39.19	54.57	63,859
Tegna	27.06	3.95	21.35	38.47	64,017
Twenty-First Century Fox Class A	29.39	2.62	24.84	34.68	64.336

on the other hand, it can find best solution without searching the whole search space.

At the end of related work, it should be mentioned that our proposed approach has two advantages over the previous methods. The first one is that the reinforcement learning agent can find the best model parameters rather than using the fixed parameters to maximize the objective function, i.e., the Sortino ratio. Another advantage of using reinforcement learning is that it can model the stocks fluctuations as a dynamic behavior of the environment and therefore find the best model parameters with respect to environment changes.

3 Data

Intraday price data are used to implement pairs trading strategy in this research. More specifically, since the parameter estimation and spread equation calculation would finally result from comparing the corresponding price trends of two assets, these two time series would be converted to discrete and comparable series through transforming them into one-minute intervals and applying the close price in each minute as its representative. Therefore, these discrete series would be used.

Data are obtained from FactSet Research Systems, Inc. [FactSet], which consist of intraday data of some stocks' price such as Facebook (FB), Google (GOOG), Amazon (AMZN), Twitter (TWTR), Alibaba (BABA), General Motors (GM), Ford Motor Company (F), Center Point Energy (CNP), NextEra Energy (NEE), Dow Chemical (DOW), LyondellBasell (LYB), Adobe Systems Inc (ADBE), Autodesk Inc (ADSK), Amphenol Corp A (APH), Boeing Company (BA), Johnson & Johnson (JNJ), Medtronic plc (MDT), Bank of America Corp (BAC), The Bank of New York Mellon Corp (BK), Helmerich & Payne (HP), Hess Corporation (HES), The Hershey Company (HSY), Tyson Foods (TSN), Tegna (TGNA), and Twenty-First Century Fox Class A (FOXA).

Although these seven time series have historically tracked each other, for model estimation, we have focused only on seven months of data from June 2015 to January 2016. Table 1 shows some summary statistics of the entire sample.

One of the most commonly used pairs is General Motors (GM) and Ford Motor Company (F). We collected 60,865 intraday prices of this pair. As shown in Fig. 1, their prices moved together.

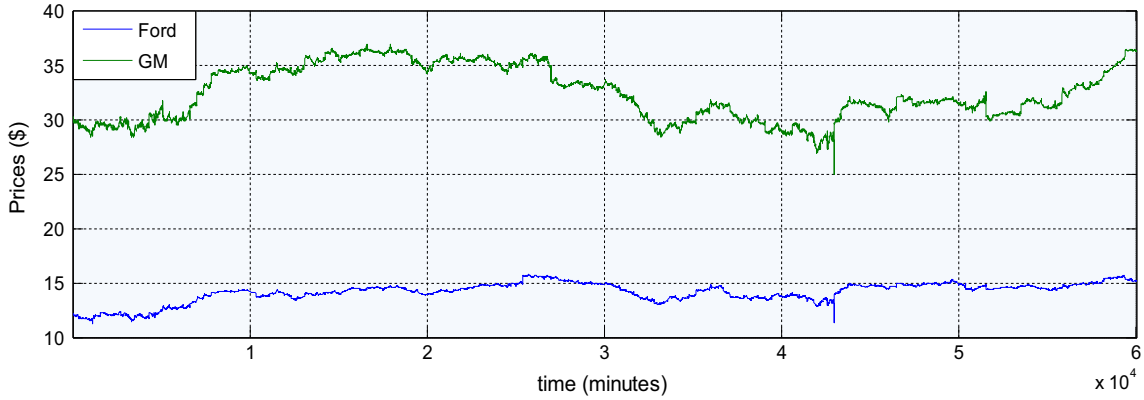

Fig. 1 Intraday prices of Ford and GeneralMotors

4 Methodology

4.1 Cointegration and error correction model

Cointegration was first stated in the primary paper by Granger (1981) as a standard tool in statistical methods in order to analyze the economic problems. To be more precise, cointegration is a general concept for describing a stationary relationship between non-stationary variables. According to a general rule, if two or more time series are non-stationary, the time series resulting from their linear combination might also be non-stationary. Assume that x_t and y_t are two non-stationary time series and the integration rank of them are d and e, respectively:

$$x_t \sim I(d), \quad y_t \sim I(e), \quad z_t = \alpha x_t + \beta y_t \tag{1}$$

Therefore, if $e > d$, then there will be $z_t \sim I(e)$. After subtracting z_t for e times, the stationary time series will be obtained as follows:

$$\Delta^e z_t = \alpha \Delta^e x_t + \beta \Delta^e y_t \tag{2}$$

The exceptional cointegration is general in this rule. For instance, in the following regression equation:

$$y_t = u_t + \beta x_t \tag{3}$$

Assume that $x_t \sim I(1)$, $y_t \sim I(1)$; therefore, the disturbance term would be equal to $u_t = y_t - \beta x_t$ which is non-stationary and $I(1)$, according to the above-mentioned discussion. If assumed that the disturbance term is *iid*, the assumption of being stationary can easily be refuted and therefore the statistical inferences will no longer be valid. Moreover, the relevant regression is called a spurious regression because the basic assumptions of its validity have been refuted. However, if there was a value for β to make $y_t - \beta x_t$ stationary, then x_t and y_t are called cointegrated variables. It means that although the above-mentioned variables are non-stationary, a linear combination of them using a specific value of β is stationary. Consequently, the aforementioned regression equation would be spurious, and β could be estimated in a correct way. Additionally, if u_t is stationary, the above-mentioned equation is referred to as cointegrated, and they do not get separated from each other in a long-run equilibrium relationship.

Furthermore, there is a close relationship between cointegration and error correction models (ECM). Such a relationship was clarified through Granger's representation theorem. Considering a set of economic variables which were in a long-run equilibrium according to the following relationship, Engle and Granger started their analysis:

$$\beta_1 x_{1t} + \beta_2 x_{2t} + \cdots + \beta_n x_{nt} = 0 \tag{4}$$

Assuming that β and x_t indicate the vectors of $(\beta_1, \beta_2, \ldots \beta_n)$ and $(x_{1t}, x_{2t}, \ldots x_{nt})'$, respectively, the above-mentioned system will be in equilibrium if $\beta x_t = 0$.e_t is the deviation from the long-run equilibrium. It is also called equilibrium error as follows:

$$e_t = \beta x_t \tag{5}$$

If the equilibrium error of a process is stationary, then the above-mentioned equation is significant. Engle and Granger (1987) defined cointegration as follows:

1. The elements of the cointegrated vector x_t should be of order d;
2. A vector like $\beta = (\beta_1, \beta_2, \ldots \beta_n)$ should exist so that for each $b > 0$, the linear combination $\beta x_t = \beta_1 x_{1t} + \beta_2 x_{2t} + \cdots + \beta_n x_{nt}$ is cointegrated of the order (d-b).

Therefore, vector β is called the cointegrated vector.

One of the main characteristics of a cointegrated vector is that its temporal trend is influenced by the deviations result-

ing from the long-run equilibrium. So, a system returns to the long-run equilibrium if the minimal changes in some variables are in the reverse direction of imbalance. This will not occur unless with a dynamic model which is in fact an error correction process. In an error correction model, short-run changes in the system variables occur based on the deviation of the system from the long-run equilibrium.

In a simple VAR model:

$$y_t = a_{11} y_{t-1} + a_{12} z_{t-1} + \varepsilon_{yt} \tag{6}$$

$$z_t = a_{21} y_{t-1} + a_{12} z_{t-1} + \varepsilon_{zt} \tag{7}$$

in which ε_{yt} and ε_{zt} are white noise disturbance terms which may be correlated with one another. For the sake of simplicity, it is assumed that the models do not have constant components. Using the interrupt operand, the above-mentioned equations are written as follows:

$$(1 - a_{11}L)\, y_t - a_{12} L z_t = \varepsilon_{yt} \tag{8}$$

$$-a_{21} L y_t + (1 - a_{22}L)\, z_t = \varepsilon_{zt} \tag{9}$$

This system is rewritten in a matrix form to calculate the values of y_t and z_t out of the above-mentioned equations, so:

$$\begin{bmatrix} (1 - a_{11}L) & -a_{12}L \\ -a_{21}L & (1 - a_{22}L) \end{bmatrix} \begin{bmatrix} y_t \\ z_t \end{bmatrix} = \begin{bmatrix} \varepsilon_{yt} \\ \varepsilon_{zt} \end{bmatrix} \tag{10}$$

According to Cramer's rule or reverse matrixes, the answers to y_t and z_t are as follows:

$$y_t = \frac{(1 - a_{22}L)\, \varepsilon_{yt} + a_{12} L \varepsilon_{zt}}{(1 - a_{11}L)\,(1 - a_{22}L) - a_{12} a_{21} L^2} \tag{11}$$

$$z_t = \frac{a_{21} L \varepsilon_{yt} + (1 - a_{11}L)\, \varepsilon_{zt}}{(1 - a_{11}L)\,(1 - a_{22}L) - a_{12} a_{21} L^2} \tag{12}$$

In fact, the first-order bivariate system presented in the above-mentioned equations was divided into two subtractive one-variable equations of the second order according to what was mentioned in the previous part. The notable point is that the characteristic roots of both equations are the same and equal to $(1 - a_{11}L)\,(1 - a_{22}L) - a_{12}a_{21}L^2$.

If the value of this characteristic root is put to zero, the roots of the reverse characteristic equation will be obtained based on the interrupting operand (L). Therefore, the parameter λ is defined as $\lambda = \frac{1}{L}$, and the characteristic equation is written as follows:

$$\lambda^2 - (a_{11} + a_{22})\,\lambda + (a_{11}a_{22} - a_{12}a_{21}) = 0 \tag{13}$$

Since both variables have the same characteristic roots, the roots of the equation indicate the characteristics of temporal trend for both variables.

It is necessary for one of the two characteristic roots, which are (λ_1, λ_2), to be equal to one, and the absolute value of the other one should be less than one. Therefore, $\{y_t\}$ and $\{z_t\}$ will be convergent or equaled to CI(1,1). Under these conditions, the variables have the same random trends, and their first subtraction will be stationary. Continuing the computational process and normalizing the equations based on y_t, for instance, if $\lambda_1 = 1$ and $|\lambda_2| < 1$, an error correction model will be obtained as follows:

$$\Delta y_t = a_y\,(y_{t-1} - \beta z_{t-1}) + \varepsilon_{yt} \tag{14}$$

$$\Delta z_t = a_z\,(y_{t-1} - \beta z_{t-1}) + \varepsilon_{zt} \tag{15}$$

If $(y_{t-1} - \beta z_{t-1})$ is in the above-mentioned equation, then y_t and z_t are influenced only by the ε_{zt} and ε_{yt}. This requires that $\beta \neq 0$ and that at least one of the speed equilibrium parameters a_y or a_z must be nonzero.

Generally, if the necessary condition for the relationship CI (1, 1) among the variables is met, then an error correction model shall be definable for these variables. Therefore, the fact that there is an error correction model for the $I(1)$ variables points out that a cointegration relationship exists among these variables. This result is actually consistent with Granger's representation theorem according to which the error correction and cointegration model would have the same representation for each set of $I(1)$ variables.

4.2 Cointegration test

There are many methods for conducting the cointegration test for selecting the appropriate pair for the pairs trading method such as Engle–Granger's method (1987) and Johansen's method (1988). Johansen's test was used to examine the cointegration and to identify the pair assets in this paper. To put it simply, Johansen's method (1988) is actually the generalization of Dickey Fuller test to the multi-variable version which uses the one-step method based on the relationship between the matrix and its characteristic roots to check cointegration. One advantage of this method is its fewer errors in comparison with the two-step method introduced by Engle–Granger (1987). Moreover, Engle–Granger's method is highly dependent on the type and the number of variables. However, these flaws are all resolved in Johansen's method which estimates the cointegration relationships in the form of VECM using the maximum likelihood method under the assumptions about the trend, parameters, and the number of vectors indicated with r. Johansen's method used these to run likelihood tests. Conducting the trace-statistic test for vector r, this method investigated the number of cointegration vectors for $r = 0, 1, \ldots, k-1$. Based on the log-likelihood ratio $\ln[L_{\max}(r)/L_{\max}(k)]$, this test actually investigates the trace of a diagonal matrix including normalized eigenvalues. It also compares the null hypothesis (with cointegration

order r) to the opposite hypothesis (with cointegration order k). Now, considering trace-statistic and p values, the existence of cointegration is investigated between the pair asset. Finally, Johansen's test generates the values of normalized cointegration coefficients in order to adopt trading strategies.

4.3 Pairs trading strategy: a cointegration approach

As stated, long-run economic relationships are estimated and analyzed among the assets in cointegration. In fact, the main idea of cointegration analysis is that although many of economic time series are non-stationary with an ascending or descending random trend, it may be possible that a linear combination of these variables will always be stationary and without a random trend in the long run. In the following, the pairs trading strategy with cointegration approach is presented in three main sections.

4.3.1 Pairs selection

Generally, there are $\binom{n}{2}$ possible states to select each pair stock out of n stocks in the market. Certainly, it is not a proper option to check and run cointegration tests on all the possible states. Applying the filters in accordance with the basic analyses and the information pertaining to previous procedures, the most appropriate stocks were selected to implement our proposed strategy. Since the probability of a cointegration relationship between two stocks of a common industry is higher, the candidate pair stocks were selected out of the available stocks of an industry through applying some criteria such as liquidity, turnover rates, and the size of deals. In this paper, based on the mentioned criteria, the fourteen appropriate pairs are used in order to test the performance of pairs trading strategies.

4.3.2 Estimating parameters through VECM

After selecting the candidate pair stocks in the previous stage, the error correction vector equations were estimated to predict the parameters and spread equation. According to Granger's theorem, there should be a short-run relationship corresponding to each long-run economic relationship in the form of an error correction model (ECM) in order to achieve the long-run equilibrium. In fact, if there is no mechanism to balance the variables in case of imbalance compared with the long-run equilibrium relationship, such a relationship will not hold among the variables in the long-run. Therefore, as indicated, cointegration would require ECM. One of the main characteristics of cointegration vectors is that their temporal trend is influenced by the deviations caused by the long-run equilibrium. Therefore, a system returns to the long-run equilibrium if the minimal changes in some variables act in the opposite direction with respect to the introduced imbalance.

The long-run equilibrium relationships for two variables (assets or stocks) of A and B are as follows:

$$\Delta A_t = \alpha_A \left(A_{t-1} - \beta A_{t-1} + \rho_A\right) + \cdots + \varepsilon_{At} \tag{16}$$

$$\Delta B_t = \alpha_B \left(B_{t-1} - \beta B_{t-1} + \rho_B\right) + \cdots + \varepsilon_{Bt} \tag{17}$$

The expressions inside the parentheses refer to the long-run equilibrium relationships called spread. It can be rewritten as a mean (μ) and a white noise expression (ε_t) as follows:

$$A_{t-1} - \beta B_{t-1} + \rho = \mu + \varepsilon_t \tag{18}$$

Now, in terms of cointegration characteristics, the spread has a constant mean during the time period. Therefore, the standardized value of spread can be defined so that trading positions can be adopted based on it.

$$\text{Indicator} = \frac{\text{spread} - \text{mean (spread)}}{\text{STD (spread)}} \tag{19}$$

4.3.3 Designing and implementing pairs trading strategies

One of the most important parts in pairs trading is to design the pairs trading strategy and to determine the optimal values of its parameters.

1. Selecting the appropriate time window in order to re-estimate the parameters: in fact, selecting proper sequences to run cointegration tests, estimating the cointegration parameters and coefficients, and re-estimating the spread equation.
2. Trading window: If there is cointegration, its coefficients, parameters, and spread equation will be true throughout the trading window period.
3. Trading thresholds (Δ): Thresholds above and below the spread mean in order to issue trading signals and take the positions.
4. Stop-loss: Thresholds on two sides of the spread mean and wider than Δ thresholds, used to close the positions at the time of loss and prevent further losses. One of the risks of pairs trading strategies is that the spread is driven away from its long-run mean in the long run. This indicates the importance of determining the optimum for the stop-loss position.

In the previous studies, specific constant values were often attributed to these parameters, whereas the reinforcement learning was used in this research to select the optimal values for these parameters to maximize the Sortino ratio. Using RL instead of allocating constant values to each parameter for designing the trades, and applying its experiences from

preprocesses and previous information, the agent selects the optimal parameters according to its experiences and discoveries each time the portfolio is open or closed. It is obvious that these four parameters can be dynamically initialized. The explanations on how to apply the reinforcement learning algorithm and its performance are presented in the next section.

To implement the strategy with spread deviation from its long-run equilibrium value, if the spread meets one of the high or low Δ thresholds pertaining to that time period, it opens the trading positions for its asset portfolios (two assets here). After occurring one of the two states of modifying the deviation and spread return to its mean or declining the stop-loss through taking reverse positions, the trading position is closed. Now, the information and value function of the algorithm are updated again, and the algorithm decides to initialize the aforementioned four parameters. As pointed out, the Sortino ratio was used as the target function and a scale to evaluate the algorithm performance with respect to simultaneous return and risk.

4.4 Reinforcement learning (modeling based on the N-Arm Bandit problem)

Reinforcement learning consists of two main components, namely the agent and the environment. The agent lives in the environment and updates its experiences through receiving feedback from the environment. Officially, the agent is in the state of s at each moment in reinforcement learning. Selecting an action (a) out of its actions space and doing it, the agent moves to the next state (s') and receives the reward (r) from the environment. After that, based on the acquired reward, it updates the experience of being in the state s and doing the action a. Updating the agent's experience is to estimate the value of the desired state and the action which has been done in that state (estimating the value of state–action). The following relation is used to estimate the value of state–action:

$$\text{Acquired reward} \times \text{learning rate} + \text{previous estimation} \rightarrow \text{new estimation}$$

The learning rate ($\alpha \in [0, 1]$) decreases as time increases. At the beginning of agent's life, the value of α is close to one because the agent does not have any experience at first. As time passes, since the agent receives feedback from the environment at each state change, it can consider the estimated value of its state–action as more important and decrease the effect of feedback from the environment for updating of its new estimations. When the learning rate is zero, the agent selects the best action in each state only through estimating the value of its state–action. Therefore, the agent can identify the behavior of the environment during its life and makes the best decision to maximize its reward in different states.

Each action is taken by the agent in accordance with a learning policy in each stage. The learning policy actually specifies the way the agent behaves through the time. Given the objective of agent and the environmental assumptions in which the agent is placed, different policies can be used. For instance, the ϵ-greedy policy can be mentioned. In this policy, the agent behaves greedily as much as ϵ and randomly as much as $1 - \epsilon$. In other words, ϵ is selected out of the practical decisions cases which had the highest value to the agent. On the other hand, $1-\epsilon$ is selected out of the decisions cases in which an action is selected by the agent at random and independent of its value. As the agent's knowledge increases, the value of ϵ can gradually decrease and drive the agent toward the selection of more promising actions which can be selected through experience of living in the environment. The pseudo-code of a reinforcement learning agent who lives by the ϵ-greedy policy is presented in Algorithm 1:

Initialize $Q(s, a)$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}(s)$, arbitrarily, and $Q(terminal - state, .) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from $Q(e.g., \epsilon - greedy)$
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from $Q(e.g., \epsilon - greedy)$
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Algorithm 1—Pseudo-code of a reinforcement leaning agent who behaves by the ϵ-greedy policy.

One of the well-known problems in the reinforcement learning is the problem of n-arm bandit. It is actually a device consisting of N arms. If each arm is pulled, a score as a reward is displayed and given by the device to the player. Each player can pull a limited number of the arms and receive reward as much as the total scores. This problem is used in the reinforcement learning in order to design an acquisitive agent. In other words, modeling the problem of n-arm bandit in an environment, an agent which can gain the highest possible score through playing with the device is designed. Therefore, in each stage, the agent decides to pull an arm and receive a score from the device. The state space includes only one state in this problem, and the actions space includes all the arms. After selecting and pulling an arm in each stage and receiving the score, the agent returns to the state in which it can select another arm to pull and receive score. Therefore, this problem has only one state called single state. In each stage, the agent

has to select an arm which leads it to a higher score. Since the score which is acquired after pulling each arm would follow a particular distribution which is not known to the agent, the agent tries to select an arm which gives a higher average score through discovering the environment. The learning rate decreases as the frequency increases, and the importance of selecting an arm which has the highest estimation value so far increases.

The problem of n-arm bandit can give ideas in order to maximize the profit obtained in pairs trading of a reinforcement learning agent. Since the behavioral change in two stocks is influenced by many uncertain factors in time, and modeling these factors as a simple noise will not be very effective, it appears that designing a reinforcement learning agent which can model different uncertain factors well and independent of the type of its impact will be more efficient than traditional methods [Sutton book].

Determining the proper value for estimation window, trading window, trading thresholds (Δ) and stop-loss parameter in pairs trading, the Sortino ratio resulting from the trades can be maximized. The values which all four parameters accept are in a limited interval. The parameters of estimation window and trading window accept the discrete values with a one-minute precision. On the other hand, the parameters of delta and stop-loss have continuous values. If we make the values of delta and stop-loss parameters discrete in a way, we can consider the problem of pairs trading as like as the problem of n-arm bandit. Therefore, we can design an agent which is able to find the appropriate value for the above-mentioned parameters after cointegration testing the pair equations of two desired stocks which were done in the past.

In this case, when a portfolio is open, the agent should select an appropriate value for the parameters. After the portfolio is closed, the agent should consider the obtained Sortino ratio as the reward. Therefore, the state of agent is the opening time of the portfolios which all constitute one state indeed. The agent's actions space includes four parameters which have been mentioned. The agent should select a value for each of these parameters through estimating the value of its state–action at the time of opening the portfolio. Using the acquired Sortino ratio, the agent updates the estimation of the values of its state–action after the portfolio is closed. In other words, like the problem of n-arm bandit, the agent should select a value for each of these four parameters when the portfolio is open, and the Sortino ratio acquiring from this selection is updated when the portfolio is closed. So, the agent returns to its initial state and waits for another portfolio to open so that it can select another four values. As previously mentioned, the values of the first two parameters were discrete, whereas the values of the second two parameters were continuous. Since the actions state is discrete in the problem of n-arm bandit, the values of delta and stop-loss parameters can be made discrete through considering a precision (for instance 0.5 of unit). Now, the values of all parameters are discrete. Each permutation of the values of all parameters would be considered an action (In fact, each one represent an arm in the problem of n-arm bandit.). Modeling the problem of pair equations as like as the problem of n-arm bandit, the agent is designed and simulated. As pointed out before, the agent should adopt a policy to select the optimal action in each decision. Considering the fact that the actions space is discrete, and it is preferable that the designed agent should have a good performance despite being simple, the $\epsilon\text{-}greedy$ policy is used as the decision-making policy. The results indicated the effectiveness of the method proposed to design the reinforcement learning agent in comparison with the previous methods. The algorithm of our proposed method is as follows:

Algorithm 1 (RL Pairs Trading)

Input: *Stock A, Stock B*

1. *Initialization*:
   a. $action\ space_{n\times 4} \leftarrow [\delta_{n\times 1},\ stop\text{-}loss_{n\times 1},\ estimation\ window_{n\times 1},\ trading\ window_{n\times 1}]$
   b. $number\ of\ iteration \leftarrow max\ number\ of\ iteration$
2. **For** $episode \leftarrow 1\ to\ number\ of\ iteration$
3. $action \leftarrow choose\ action\ based\ on\ the\ epsilon\ greedy\ policy$
4. $reward \leftarrow perform\ action\ (action, A, B)$
5. $action\ update\ (action, reward)$
6. **End**
7. **Return** $Sortino\ ratio$

Output: *Sortino Ratio*

In this algorithm, the actions space is first initialized in accordance with what was pointed out. This space includes four parameters of estimation window, trading window, trading thresholds (Δ) value, and stop-loss value. Then, the enough number of iteration is selected to train the agent. After that, for each episode, selecting one value out of four above-mentioned parameters as an action of the agent is done in accordance with the $\epsilon\text{-}greedy$ policy. According to the methodology proposed in Part 4, then the value of Sortino ratio is calculated as the reward using *perform action* function. It is returned to the agent as the feedback from the environment. Using the *action update* function and the value of reward acquired in the previous step, then the agent's experience of state–action value of each of the four parameters is updated. Finally, the overall Sortino ratio acquired after evaluating all the input data is returned as the output.

5 Numerical results

This section emphasizes on testing the performance of our proposed method. Also, the results of our proposed method

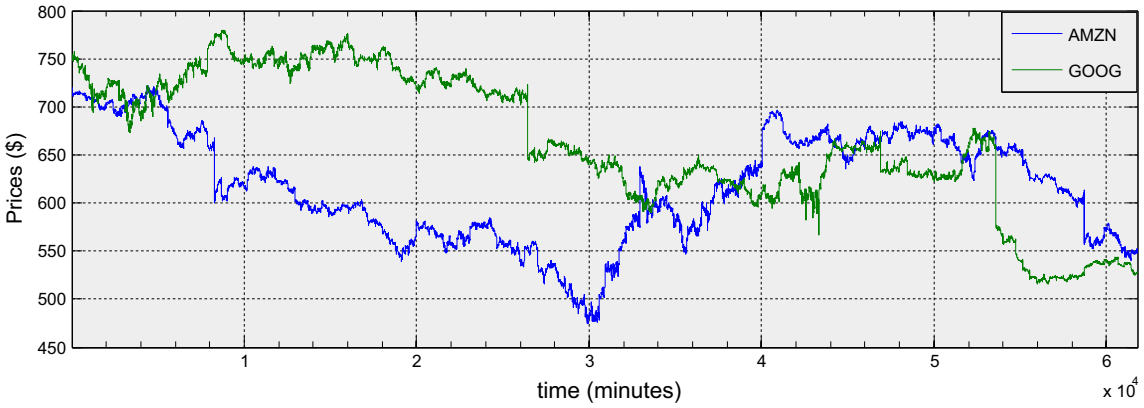

Fig. 2 Intraday prices of Amazon and Google stocks

are compared with the results acquiring from the constant parameters method (CPM). Furthermore, the simulation parameters are described at first. Then, data and test cases are presented and the robustness test is performed. Finally, discussion and analysis of results are explained.

5.1 Simulation parameters

In this section, the simulation parameters are described. As previously noted, our main contribution is to propose a new strategy for pairs trading using a reinforcement learning agent. Also, in our simulation, the Sortino ratio is considered as a reward function.

Our simulations has been done in MATLAB version 2013a environment running on a machine with Corei5 2.4Ghz and 4GB RAM. The value of four parameters including: backward window, trading window, spread and stop-loss varies as the following:

- Estimation window varies between [60,600] with the step of 5 min.
- Trading window varies between [5,120] with the step of 5 min.
- Trading thresholds varies between [0,3] with the step of 0.5.
- Stop-loss varies between [0,5] with the step of 0.5.

Considering the concept of the reinforcement algorithm which needs to learn to choose suitable parameters in pairs trading strategy, we have selected 75 % of data as in-sample for the training phase and the rest of it as out-sample data for the testing phase. The training phase is iterated to 100 times and we set $\alpha = 1, \epsilon = 1$. In each iteration of the training phase, a value is chosen for each of four parameters based on the ϵ-$greedy$ policy and the value of current state–action is updated. After that, in testing phase, the agent chooses parameters for the pairs trading strategy and $\alpha = 0.3, \epsilon = 0.3$ are set.

5.2 Data and testcase

The fourteen appropriate data are considered as intraday data and located between June 2015 and January 2016. These data are adapted from various companies which are FB-GOOG, BABA-AMZN, FB-TWTR, GM-F, AMZN-GOOG, CNP-NEE, DOW-LYB, ADBE-ADSK, APH-BA, JNJ-MDT, BAC-BK, HP-HES, HSY-TSN and TGNA-FOXA. Trending of Amazon–Google's intraday prices is shown in Fig. 2 in which time = 0 indicates the time of the first data which is equal to 9th June.

As it can be seen in Fig. 2, the prices follow each other.

5.3 Simulations results

We initially chose 420-min window for test and estimation and also 30-min window for trading. Then, we have ran a grid search algorithm on both windows with the 5 and 20 min as the steps for the trading window and test and estimation window, respectively.

The results of the initial, best, and worst window sizes in regard to the return are presented in Table 2.

Based on the above table, we choose the best parameters, i.e., Best column, for the CPM.

The results of the Amazon-Google pairs of the in-sample data for the CPM and the Reinforcement Learning Method (RLM) are shown in Figs. 3 and 4, respectively. It should be mentioned that the estimation window, the trading window, the trading threshold, and the stop-loss were set to 420 min, 60 min, 2, and 3, respectively, in the CPM. Note that, MAR = 0.02 is chosen as the risk-free rate of the return on the numerator of the Sortino ratio which is calculated in each

Table 2 Window size optimization results in regards to return

	Initial	Best	Worst
Test and estimation window size (min)	420	380	240
Trading window size (min)	30	80	20
Return (%)	9.50	31.6	−83.55
Volatility	0.55	0.64	0.85
Sortino ratio	0.13	0.15	−1.47
Number of trades	390	265	563

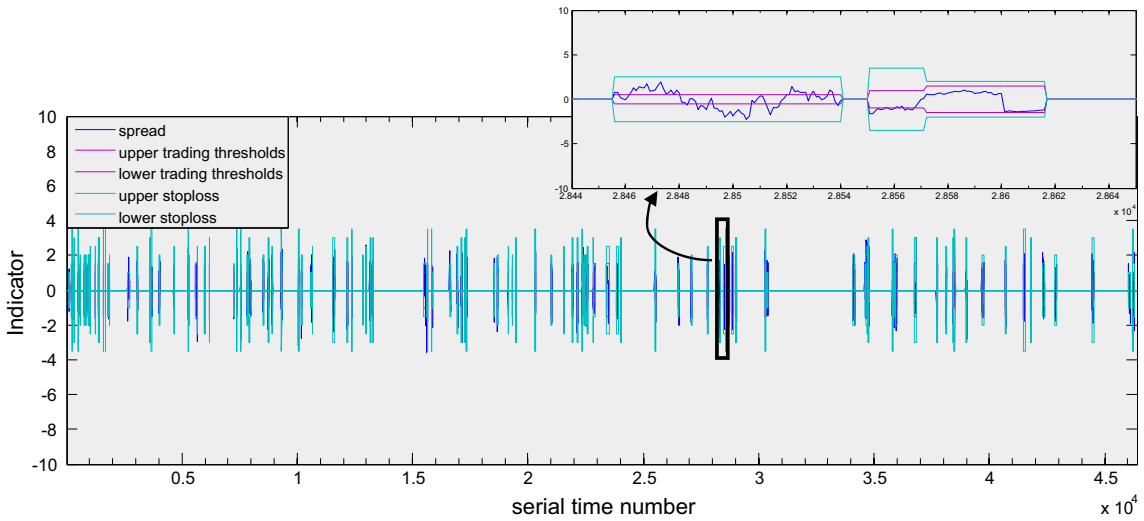

Fig. 3 Indicator and positions on the in-sample data of Amazon–Google pair which have been calculated by the proposed method (RLM)

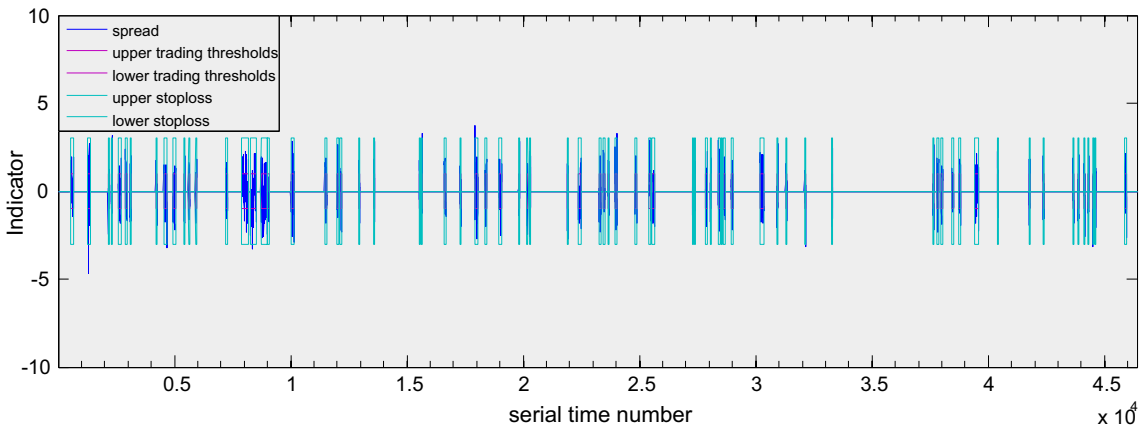

Fig. 4 Indicator and positions on the in-sample data of Amazon–Google pair which have been calculated by the constant parameters method (CPM)

iteration of the algorithm. Also, in Figs. 5 and 6, the results of the out-sample data are shown.

Using the reinforcement learning for modeling dynamic parameters, the estimation window, the trading window, the trading thresholds and stop-loss parameters are varied in each episode to maximize the Sortino ratio.

The summary of the results are shown in Table 3.

Table 3 shows that the proposed method overperforms the CPM in which the return values of pairs trading in RLM over in-sample and out-sample data are much more than the CPM. Moreover, the Sortino ratios over in-sample data are 1.46 and 0.27 for RLM and CPM, respectively. In out-sample

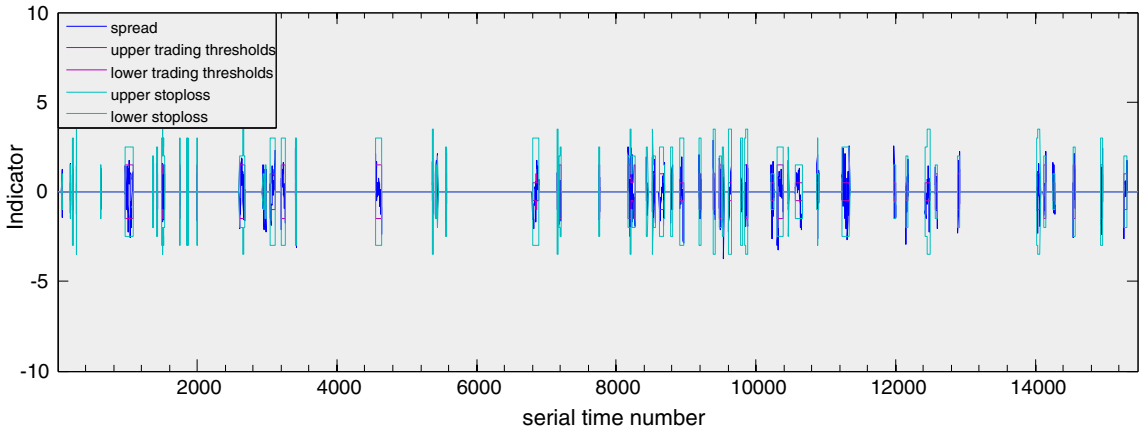

Fig. 5 Indicator and positions on the out-sample data of Amazon–Google pair which have been calculated by the proposed method (RLM)

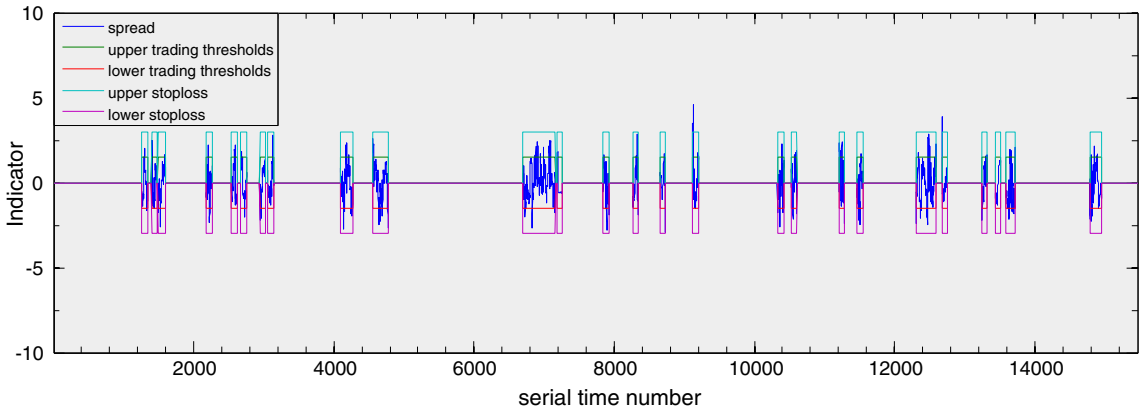

Fig. 6 Indicator and positions on the out-sample data of Amazon–Google pair which have been calculated by the constant parameters method (CPM)

Table 3 Simulation results on Amazon–Google pair over the in-sample and out-sample data for both the proposed method (RLM) and the constant parameters method (CPM)

	CPM		RLM	
Test and estimation window size (min)	380		Dynamic [60,600]	
Trading window size (min)	80		Dynamic [5,120]	
Δ (standard deviation)	1.5		Dynamic [0.5,3]	
Stop-loss (standard deviation)	3		Dynamic [1,5]	
	In-sample	Out-sample	In-sample	Out-sample
Return (%)	15.8	11.3	91.5	46.6
Downside volatility	0.51	0.18	0.61	0.12
Annualized return (%)	31.6	67.8	183	279.6
Sortino ratio	0.27	0.52	1.46	3.72
Number of trades	265	180	586	388
Average return per trade (%)	0.06	0.06	0.16	0.12
Average time per trade (min)	185.4	91	83.8	42.2

data, the Sortino ratios of RLM and CPM are 3.72 and 0.52, respectively. Furthermore, the number of trades in RLM is higher than the CPM. More specifically, the number of trades on RLM and CPM are 388 and 180, respectively. On the other hand, the average return values per trade on RLM and CPM are 0.12 and 0.06 %, respectively. Moreover an agent can

explore the environment and learn from past experiences, and it can select the best model parameters including estimation windows, the trading thresholds, and stop-loss to maximize the Sortino ratio. Consequently, the higher number of trading positions and less average time per trade on RLM have been acquired compared to CPM.

Note that the proposed method is iterated up to 100 times in the training phase which means the agent experiences only 100 states from 100 K possible states which acquire regarding the resolution of each parameter in the initiation step of the RLM. It is obvious that increasing the number of iteration allows the agent to experience the more number of all possible states which can lead to better training of the agent.

5.4 Robustness test

The robustness test shows the degree of a system which is able to generate the correct output in the case of different inputs and conditions. Furthermore, the goal of the robustness test is to develop test cases to examine the robustness of a system. To achieve this goal, we consider fourteen appropriate pairs to compare RLM with the CPM. The results are shown in Tables 4, 5, and 6.

Tables 4, 5, and 6 show that the RLM overperforms significantly much better than the CPM in term of the Sortino ratio and the average return value. Therefore, based on the results, it can be concluded that our proposed method (RLM) is completely robust in the presence of different inputs and conditions. Moreover, to examine the RLM statistically, the T-test is performed on the return values of the RLM and the CPM. It should be noted that the return value of CPM is a constant value in each run of it. The results of the T-test over out-sample which has been ran for 15 times are summarized in Table 7.

Table 7 shows that the RLM achieved significantly higher results than the CPM by the confidence level of 95 %.

6 Conclusion

The prior researches consider constant parameters including the estimation window, the trading window, the trading thresholds, and stop-loss for pairs trading strategies. In this paper, we proposed a new method to determine the values of four parameters dynamically using the reinforcement learning to maximize the Sortino ratio. In fact, using the reinforcement learning, the agent experiences the long-run relations between pairs prices and it can choose a better value for four parameters to maximize the Sortino ratio. The results of our simulations over fourteen pairs showed the effectiveness of our proposed method (RLM) such that it overperforms significantly much better than the method with constant para-

Table 4 Results of the proposed method (RLM) and the constant parameters method (CPM) on the different pairs including FB-GOOG, AMZN-GOOG, FB-TWTR, BABA-AMZN, and F-GM

	FB-GOOG				AMZN-GOOG				FB-TWTR				BABA-AMZN				F-GM			
	CMP		RLM		CMP		RLM		CMP		RLM		CMP		RLM		CMP		RLM	
	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
Return (%)	6.1	5.8	83.6	41	15.8	11.3	91.5	46.6	17.4	8.1	56	38.7	10.5	−7.8	108	66	35.2	−18.5	115.6	58.8
Downside volatility	0.15	0.28	0.31	0.26	0.51	0.18	0.61	0.12	0.22	0.36	0.31	0.17	0.61	0.27	0.11	0.18	0.59	0.87	0.19	0.77
Annualized return (%)	12.2	34.8	167.2	246	31.6	67.8	183	279.6	34.8	48.6	112	232.2	21	−46.8	216	396	70.4	−111	231.2	352.8
Sortino ratio	0.27	0.14	2.63	1.5	0.27	0.52	1.46	3.72	0.7	0.17	1.74	2.16	0.14	−0.36	9.8	3.55	0.56	−0.24	5.98	0.74
Number of trades	305	162	582	199	265	180	586	388	153	96	293	211	110	86	332	154	216	105	366	280
Average return per trade (%)	0.02	0.03	0.14	0.2	0.06	0.06	0.16	0.12	0.11	0.08	0.19	0.18	0.09	−0.09	0.32	0.43	0.16	−0.18	0.32	0.21
Average time per trade (min)	161.1	101.1	84.4	82.3	185.4	91	83.8	42.2	321.2	170.6	167.7	77.6	446.7	190.5	148	106.4	227.5	156	134	58.5

Table 5 Results of the proposed method (RLM) and the constant parameters method (CPM) on the different pairs including CNP-NEE, DOW-LYB, ADBE-ADSK, AHP-BA and, JNJ-MDT

	CNP-NEE				DOW-LYB				ADBE-ADSK				AHP-BA				JNJ-MDT			
	CMP		RLM		CMP		RLM		CMP		RLM		CMP		RLM		CMP		RLM	
	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
Return (%)	−10.7	6.3	70	62	18	12.6	59	53	19.2	−16.4	62.8	31.8	23.8	13.3	142	98.6	30.6	−23	100.5	43.5
Downside volatility	0.98	0.5	0.22	0.41	0.89	0.2	0.44	0.07	0.23	0.47	0.31	0.12	0.61	0.72	0.85	0.18	0.66	0.31	0.19	0.1
Annualized return (%)	−64.2	12.6	140	372	36	75.6	118	318	38.4	−98.4	125.6	190.8	47.6	79.8	284	590.4	61.2	−138	201	261
Sortino ratio	−0.13	0.09	3.09	1.46	0.18	0.53	1.29	7.29	0.75	0.39	1.96	2.48	0.36	0.16	1.64	11.9	0.43	−0.8	5.18	4.15
Number of trades	315	185	614	221	380	157	596	311	114	77	358	251	144	57	459	233	221	105	652	299
Average return per trade (%)	−0.03	0.03	0.11	0.28	0.05	0.01	0.1	0.17	0.17	−0.21	0.17	0.13	0.16	0.23	0.3	0.42	0.14	−0.22	1.54	0.15
Average time per trade (min)	156	88.5	80	74.1	129.3	104.3	82.4	52.6	431	212.7	137.3	65.3	341.3	287.4	107	70.3	22.3	156	75.4	54.8

Table 6 Results of the proposed method (RLM) and the constant parameters method (CPM) on the different pairs including BAC-BK, HP-HES, HSY-TSN, and TGNA-FOXA

	BAC-BK				HP-HES				HSY-TSN				TGNA-FOXA			
	CMP		RLM		CMP		RLM		CMP		RLM		CMP		RLM	
	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
Return (%)	16.3	6	70.4	29	41	10.6	73.3	68.5	−13.6	4.75	62	31.6	21.1	8.2	86.4	66
Downside volatility	0.33	0.28	0.22	0.52	0.89	0.25	0.2	0.09	0.24	0.09	0.31	0.47	0.61	0.27	0.52	0.18
Annualized return (%)	32.6	36	140.8	174	82	63.6	146.6	411	−27.2	28.5	124	189.6	42.2	49.2	172.8	396
Sortino ratio	0.43	0.14	3.11	0.52	0.44	0.32	3.56	7.39	−0.48	0.3	1.93	0.63	0.31	0.23	1.62	3.55
Number of trades	347	185	480	201	380	188	588	311	250	108	358	184	198	119	463	155
Average return per trade (%)	0.05	0.03	0.14	0.14	0.1	0.06	0.13	0.22	−0.05	0.04	0.17	0.17	0.1	0.07	0.19	0.43
Average time per trade (min)	141.6	88.5	102.3	81.5	129.3	87.1	83.6	53.7	196.6	151.7	137.3	89	248.1	137.6	106.1	105.7

Table 7 Results of T-test over the fourteen pairs data for two methods

	CPM return (%)	Average return for RLM (%)	Mean difference (RLM-CPM) (%)
FB-GOOG	5.8	35.71	29.91***
AMZN-GOOG	11.3	33.73	22.43***
FB-TWTR	8.1	27.26	19.16***
BABA-AMZN	−7.8	40.84	48.64**
F-GM	−18.5	26.1	44.6***
CNP-NEE	6.3	36.6	30.3***
DOW-LYB	12.6	29	16.4***
ADBE-ADSK	−16.4	22.64	39.04***
AHP-BA	13.3	45.57	32.27**
JNJ-MDT	−23	26.99	49.99***
BAC-BK	6	20.33	14.33***
HP-HES	10.6	28.77	18.17***
HSY-TSN	4.75	25.15	20.4**
TGNA-FOXA	8.2	39.65	31.45***

*** Significantly different from zero at the level of 1 %
** Significantly different from zero at the level of 5 %

meters (CPM). Furthermore, we employed the robustness test and compared the mean differences between the CPM and the RLM methods in terms of the average return and the Sortino ratio. The results showed that the RLM is completely robust and it can perform significantly much better than the CPM by the confidence level of 95 %.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this article.

References

Andrade S, Di Pietro V, Seasholes M (2005) Understanding the profitability of pairs trading. Unpublished working paper, UC Berkeley, Northwestern University

Bertram W (2010) Analytic solutions for optimal statistical arbitrage trading. Phys A 389(11):2234–2243

Bogomolov T (2011) Pairs trading in the land down under. In: Finance and Corporate Governance Conference

Chiu MC, Wong HY (2015) Dynamic cointegrated pairs trading: mean-variance time-consistent strategies. J Comput Appl Math 290:516–534

de Moura CE, Pizzinga A, Zubelli J (2016) A pairs trading strategy based on linear state space models and the Kalman filter. Quant Finance 1–15. doi:10.1080/14697688.2016.1164886

Engle RF, Granger CWJ (1987) Co-integration and error correction: representation, estimation, and testing. Econometrica 55(2):251–276

Gao X, Chan L (2000) An algorithm for trading and portfolio management using Q-learning and sharpe ratio maximization. In: Proceedings of the international conference on neural information processing, pp 832–837

Gatev E, Goetzmann WN, Rouwenhorst KG (2006) Pairs trading: performance of a relative-value arbitrage rule. Rev Financ Stud 19(3):797–827

Granger CW (1981) Some properties of time series data and their use in econometric model specification. J Econ 16(1):121–130

Guo X, Zhang Q (2005) Optimal selling rules in a regime switching model. IEEE Trans Autom Control 50:1450–1455

Huang CF, Hsu CJ, Chen CC, Chang BR, Li CA (2015) An intelligent model for pairs trading using genetic algorithms. Comput Intell Neurosci 2015:16

Huck N, Afawubo K (2015) Pairs trading and selection methods: Is cointegration superior? Appl Econ 47(6):599–613

Johansen S (1988) Statistical analysis of cointegration vectors. J Econ Dyn Control 12(2):231–254

Kawasaki Y, Tachiki S, Udaka H, Hirano T (2003) A characterization of long-short trading strategies based on cointegration. In: Computational intelligence for financial engineering, 2003. Proceedings. 2003 IEEE International Conference, IEEE, pp 411–416

Lee JW, Park J, Lee J, Hong E (2007) A multiagent approach to Q-learning for daily stock trading. IEEE Trans Syst Man Cybern Part A Syst Humans 37(6):864–877

Moody J, Saffell M (2001) Learning to trade via direct reinforcement. IEEE Trans Neural Netw 12(4):875–889

Muslumov A, Yuksel A, Yuksel SA (2009) The profitability of pairs trading in an emerging market setting: evidence from the Istanbul stock exchange. Empir Econ Lett 8(5):1–6

Puspaningrum H, Lin YX, Gulati CM (2010) Finding the optimal preset boundaries for pairs trading strategy based on cointegration technique. J Stat Theory Pract 4(3):391–419

Perlin MS (2009) Evaluation of pairs-trading strategy at the Brazilian financial market. J Deriv Hedge Funds 15(2):122–136

Sutton RS, Barto AG (1998) Introduction to reinforcement learning. MIT Press, Cambridge

Tan Z, Quek C, Cheng PY (2011) Stock trading with cycles: a financial application of ANFIS and reinforcement learning. Expert Syst Appl 38(5):4741–4755

Tourin A, Yan R (2013) Dynamic pairs trading using the stochastic control approach. J Econ Dyn Control 37(10):1972–1981

Won Lee J (2001) Stock price prediction using reinforcement learning. In: Industrial electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on vol. 1, IEEE, pp 690–695

Zhang Q (2001) Stock trading: an optimal selling rule. SIAM J Control Optim 40(1):64–87

Zeng Z, Lee CG (2014) Pairs trading: optimal thresholds and profitability. Quant Finance 14(11):1881–1893