Expert Systems With Applications 130 (2019) 145–156

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning

Saud Almahdi, Steve Y. Yang*

School of Business, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, NJ 07030, USA

ARTICLE INFO

Article history:
Received 27 July 2018
Revised 5 April 2019
Accepted 6 April 2019
Available online 13 April 2019

Keywords:
Recurrent reinforcement learning
Particle swarm optimization
Optimal portfolio rebalancing
Portfolio constraint

ABSTRACT

This study extends a recurrent reinforcement portfolio allocation and rebalancing management system with complex portfolio constraints using particle swarm algorithms. In particular, we propose to use a combination of recurrent reinforcement learning (RRL) and particle swarm algorithm (PSO) with Calmar ratio for both asset allocation and constraint optimization. Using S&P100 index stocks, we show such a system with a Calmar ratio based objective function yields a better efficient frontier than the Sharpe ratio and mean-variance based portfolios. By comparing with multiple PSO based long only constrained portfolios, we propose an optimal portfolio trading system that is capable of generating both long and short signals and handling the common portfolio constraints. We further develop an adaptive RRL-PSO portfolio rebalancing decision system with a market condition stop-loss retraining mechanism, and we show that the proposed portfolio trading system outperforms the benchmarks consistently especially under high transaction cost conditions.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Asset allocation and rebalancing are considered the key building blocks of modern active portfolio management systems. Being able to dynamically switch between assets and re-allocate them would generate high returns and minimize inadvertent risks in a multiperiod type of investment. At the same time, the real world portfolio management systems need to also consider complex real world constrains on these portfolios, such as cardinality, quantity, round-lot, asset classes and pre-assignment constraints. We aim to extend the recurrent reinforcement portfolio allocation and rebalancing method in the current literature (Almahdi & Yang, 2017) and develop a constrained dynamic portfolio management system using particle swarm algorithms.

Mean-variance portfolio optimization was introduced by Markowitz (1952) where the portfolio asset selection is done by maximizing the expected return and minimizing the covariance matrix. However, this classic portfolio theory has faced with many theoretical and practical challenges, such as how to estimate expected return, how to handle a multi-period investment problem, how to reconcile multiple objectives, etc., which have prompted academics and practitioners to search for better alternative solutions. In recent decades, many meta-heuristic methods have been proposed to solve these types of problems by utilizing machine learning and artificial intelligence methods such as neural networks, genetic algorithm, reinforcement learning, etc. Particle swarm optimization (PSO) is one of the meta-heuristic methods developed by Kennedy (1995) that is based on swarm intelligence. Applying meta-huristic methods to portfolio optimization was first introduced by Chang, Meade, Beasley, and Sharaiha (2000) where the authors applied different algorithms to solve the constrained portfolio optimization problem.

In this paper, we extends the recurrent reinforcement learning (RRL) method with a statistically coherent downside risk adjusted performance objective function to simultaneously generate both buy/sell signals (Almahdi & Yang, 2017), and then we apply PSO method to solve some of the real world constrained portfolio problems generating the optimal portfolio weights under well-known practical constraints. Recurrent reinforcement learning was first introduced by Moody, Wu, Liao, and Saffell (1998) where a trading system was developed focusing on a single asset. Most recently, Almahdi and Yang (2017) introduced a more generalized recurrent reinforcement learning approach on portfolio management which is able to not only generate variable weights but also address the limitations of the existing performance objective measures in the portfolio optimization process. It thus opens the door for designing more flexible and high performing active portfolio management systems. However, as it is acknowledged by Almahdi and Yang (2017), their method can only be applied to unconstrained portfolio problems. The RRL represented

* Corresponding author.
E-mail addresses: salmahdi@stevens.edu (S. Almahdi), steve.yang@stevens.edu (S.Y. Yang).

https://doi.org/10.1016/j.eswa.2019.04.013
0957-4174/© 2019 Elsevier Ltd. All rights reserved.

in Almahdi and Yang (2017) applies the Calmar ratio as an objective function where the maximum of the objective function is found following a gradient based method to generate long short signals and variable weights for an unconstrained portfolio with pre-selected assets. This work extends the previous work by solving the real world constraints with the Calmar ratio as a fitness function for the PSO algorithm that follows a stochastic optimization approach to maximize the fitness function for asset selection and weights. The PSO has been selected for this study for the fact that unlike the RRL it can search through a wider range of solutions without the need to follow a gradient which makes it a suitable method for asset and weight selection in a wide range of assets and under any fitness function.

However, the PSO algorithm has never been applied for a multi-period constrained long/short portfolio optimization where a negative weight is allowed. That is why RRL is introduced in this paper so that long/short signals can be generated for the constrained portfolio where RRL will maximize the same performance measure. In the current literature, most of papers using PSO are focused on single-period constrained portfolio optimization which we have provided a thorough review on. To fill the gap, this study proposes a RRL-PSO approach to handle real world constraints in the portfolio optimization process.

Many studies in the current literature apply PSO to solve the constrained portfolio problem for a long only portfolio.[1] However, there are different PSO approaches in the constrained portfolio problem context where one may use the recurrent reinforcement learning method to generate long/short signals for dynamic portfolio optimization. We show that combining a PSO method with RRL gives a superior performance when it is applied to the stocks that constitute the S&P100 index by taking real world constraints into consideration. The addition of RRL will allow for a long/short constrained portfolio on a broad range of weights and asset selections. Moreover, we examine the objective function for the PSO algorithm training, and we find the Calmar ratio - an expected maximum drawdown risk adjusted performance measure yields better performance than the other risk performance measures such as Mean-Variance and Sharpe ratio in the existing literature. The constraint handling techniques suggested in this study depend highly on the selection of the fitness function and they are applied in a dynamic fashion. Different fitness functions might result in a different range of solutions that satisfy the constraints. We then propose an adaptive constrained portfolio trading system with stop-loss controls, and we show that the proposed active portfolio management system outperforms all the other benchmarks and is able to withstand high transaction costs. To the best of our knowledge, the proposed approach is the first active portfolio management system based on both RRL and PSO algorithms where both algorithms use the same objective/fitness function for the asset selection process and trading positions with different solving approaches.

The rest of the paper is organized as follows. In Section 2, we review existing work on dynamic portfolio optimization using reinforcement learning methods as well as constrained portfolio optimization with particle swam algorithms. We introduce a trading system with both RRL and PSO in Section 3. We apply the RRL-PSO portfolio rebalancing approach to the S&P100 constituent stocks to compare the effect of PSO algorithms and constraints in Section 4. Section 5 conducts a final analysis comparing the performance of the proposed portfolio management system with that of the existing approaches, and Section 6 concludes the study and identifies some future work.

[1] A long only portfolio is a portfolio that does not allow any short selling or negative weights.

2. Literature review

In this section, we review two main strands of literature related to our proposed system. First, we examine the existing work on constrained portfolio optimization problems. Secondly, we review reinforcement learning based portfolio management approaches in the current literature to set context for the proposed approach.

2.1. Constrained portfolio optimization

In real world trading and portfolio optimization applications, practitioners take into consideration some of the real world constraints such as cardinality constraint, quantity constraint, round-lot constraint, pre-assignment constraint and class constraint. The cardinality constraint limits the number of asset selection to a K number of assets in the portfolio. Quantity constraint places certain boundaries on the weight of each asset allocation. The pre-assignment constraint allows the investor to pre-select an asset in the portfolio for the investment horizon. The round-lot constraint poses a restriction on the number of shares per asset to an exact multiple of the normal trading lots. The class constraint is applied on a multi-asset class portfolios where it limits the proportion invested in each asset class. Although these constrained portfolio optimization problems are complex and hard to solve (Chang et al., 2000; Moral-Escudero, Ruiz-Torrubiano, & Suárez, 2006), there exists a number of heuristic search based approaches in the current literature to help practitioners address their specific needs. Many scholars followed the approach of meta-heuristics and evolutionary algorithms to solve the constrained portfolio problem. Methods such as the artificial bee colony was applied by Kalayci, Ertenlice, Akyer, and Aygoren (2017) where the authors explained two attempts in solving the cardinality constrained portfolio optimization. Liagkouras and Metaxiotis (2016) proposed a novel multiobjective evolutionary algorithm specially designed to solve the cardinality constrained portfolio optimization problem.

Particle swarm optimization (PSO) is a stochastic optimization method based on swarm intelligence. It was applied in a portfolio optimization context by Cura (2009) where the algorithm was applied on the cardinality constrained Markowitz model with upper and lower bounds on the weights. The method was compared with other heuristics namely genetic algorithm, tabu search and simulated annealing. Comparing with other methods, the authors concluded that PSO method does not clearly outperform other methods in terms of investment policies, but in the case of a low risk portfolio, PSO gives the best solution. Zhu, Wang, Wang, and Chen (2011) applied PSO method to solve the portfolio optimization problem with no real world constraints. Two single objective models were developed: an efficient frontier and a Sharpe ratio model. The authors solved a long only restricted portfolio optimization problem and an unrestricted portfolio with long and short weights with no real world constraints. Golmakani and Fazel (2011) solved the Markowitz portfolio problem with four constraints. Bounds on holdings, cardinality, minimum transaction lots and sector (or market/class) capitalization constraints by proposing a CBIPSO algorithm which is a combination of the binary particle swarm optimization (BPSO) and the improved particle swarm optimization (IPSO). Where the CBIPSO algorithm is based on two components: the selection of M securities out of N available securities satisfying the cardinality constraints and seeking the best positive integer investment factors for the M selected securities. The results of the research proved that the proposed method outperforms the genetic algorithm. An improved PSO was developed by Deng, Lin, and Lo (2012) for solving the cardinality constrained Markowitz problem. The improved algorithm satisfies the cardinality and quantity constraints and improves the exploration efficiency and the convergence speed. The improved PSO was compared with genetic algorithm, tabu search and simulated annealing where it outperformed all three algorithms and showed more robustness. Ni, Yin, Tian, and Zhai (2017) proposed four dynamic random population topology on the particle swarm optimization to tackle the cardinality constrained Markowitz portfolio optimization problem. The four topological choices are: dynamic random population topology based on the average degree (DRT-AD), dynamic random population topology based on the degree (DRT-D), dynamic random population topology based on the linear increasing average degree(DRT-LIAD), and dynamic random population topology based on the linear increasing degree (DRT-LID). The authors implemented the four strategies on particle swarm optimization with constriction factor (CPSO) where the results show that the DRTCPSO-LIAD and DRTCPSO-LID give the best performance. Sun, Fang, Wu, Lai, and Xu (2011) proposed a new PSO method called drift particle swarm optimization (DPSO) which is inspired by the motion in electronics to solve the multi-stage portfolio optimization problem. The mean-variance function was used as the objective function for the proposed method. DPSO is compared with PSO, GA, LOQO and CPLEX using the S&P100 equities for the test. They showed that DSPO gives the best performance where it generated better efficient frontier with the least computational cost.

2.2. Reinforcement learning for portfolio optimization

Reinforcement learning (RL) (Sutton, Barto, & Williams, 1992) is a part of machine learning that focuses on agents' learning by interacting with the environment. It is a model free algorithm that can be applied to many applications. Neuneier (1998) introduced the application of portfolio optimization by applying the RL Q-learning method. Li and Chan (2006) introduced a reward adjusted reinforcement learning to the portfolio optimization application, where the authors adjusted the reward in the RL model with a risk penalty obtained from the GARCH model. RL methods can be efficiently combined with different algorithms such as meta-heuristics which can lead to enhancing the learning performance. Chen, Mabu, Hirasawa, and Hu (2007) developed a genetic network programming with SARSA model (GNP-RL). By adding a technical index they were able deliver information to the (GNP-RL) where SARSA determines the buying and selling timings. Then their work was extended to a model and an enhancement to the method was proposed by Chen, Mabu, Shimada, and Hirasawa (2009). Moody et al. (1998) introduced the recurrent reinforcement learning (RRL) method which is an unconstrained RL algorithm that works in a model free framework and solves the cures of dimensionality problem. Many scholars added to the RRL thereafter. Zhang and Maringer (2013) introduced technical analysis, fundamental analysis and econometric analysis to RRL to enhance the long/short decisions generated by the RRL. The indicators were pre-screened by applying the genetic algorithm. Gorse (2010) compared the results from the RRL with the results from applying a genetic programming method. Gabrielsson and Johansson (2015) combined the RRL method with the Japanese candle sticks technical analysis techniques in futures trading. Almahdi and Yang (2017) introduced a coherent risk adjusted fitness measure, expected maximum drawdown, and developed a generalized recurrent reinforcement learning approach on portfolio management which is able to generate variable weights for a long/short dynamic portfolio rebalancing system. The proposed work in this study extends Almahdi and Yang (2017)'s work and addresses the constrained portfolio rebalancing issues using a PSO meta-heuristic optimization method.

3. Methodology and data

In this study, we apply RRL method using Calmar ratio with an expected maximum drawdown measure as the objective function to allocate assets and assign weights for a portfolio, and we use PSO algorithms to solve the constrained portfolio optimization problem. First we explain the RRL method for portfolio asset allocation and rebalancing. And then we introduce several PSO methods with the real world constraints imposed on the portfolio optimization approach.

3.1. Recurrent reinforcement learning

The recurrent reinforcement with Calmar ratio in Almahdi and Yang (2017) was applied to create an equally weighted long/short portfolio and a variable weights long/short portfolio with no real world constraints taken into consideration. Weights and signals were generated with every time step and a re-initialization of the training parameters were reset when a stop loss threshold is triggered. In this study, we apply the recurrent reinforcement learning in Almahdi and Yang (2017) in order to generate long/short signals for our portfolios. We then use an online learning approach to increase the convergence of the learning process and adapt to changing market conditions during live trading. During this process, the parameters are updated dynamically while trading in a moving window, and the influence of Calmar ratio is recalculated with every step of the moving window.

Calmar ratio is the objective function in our RRL model and the fitness function for all four PSO methods applied in this study where the ratio includes the expected maximum drawdown - E(MDD) - as defined by Magdon-Ismail and Atiya (2004). E(MDD) in the Calmar ratio is an MDD risk metric that measures the maximum cumulative loss from a peak to a following bottom.

$$C_T = \frac{\gamma T}{E(MDD)} \tag{1}$$

where C_T is the Calmar ratio over the time horizon T, $E(MDD)$ is the expected maximum drawdown, γ is the mean of the returns. The estimation of the Calmar ratio follows Magdon-Ismail and Atiya (2004) and is mentioned in Almahdi and Yang (2017).

We generate reinforcement learning long/short signals for our portfolio using an online learning process. The RRL method is defined in Almahdi and Yang (2017) showing the signal generation equations. As mentioned in Almahdi and Yang (2017) the RRL generates $\{1, -1\}$ indicating a long/short signals.

$$F_t = \tanh(\boldsymbol{x}_t' \boldsymbol{\theta}) \tag{2}$$

Let $F_t \in \{-1, 1\}$ be the trading signal with only two values. The *tanh* function in Eq. (2) is a nonlinear function that allows for F_t to take any values between −1,1. For $F_t > 0$, the investor would take a long position, and we set $F_t = 1$. For $F_t < 0$, the investor would then take a short position, and we set $F_t = -1$. θ are the parameters that we want to train $\theta \in \Re^{M+2}$ where M is the time series moving window step in this study $M = 2$. x_t is a vector where $x_t = [1; r_t \ldots r_{t-M}; F_{t-1}]$, r_t is the log return $r_t = log(price_t) - log(price_{t-1})$ (Almahdi & Yang, 2017). We calculate the return at time t in Eq. (3):

$$R_t = \mu * [F_{t-1} \cdot r_t - \delta |F_t - F_{t-1}|] \tag{3}$$

where μ is the number of shares that is a constant number. δ is the transaction cost and it is also a constant.

3.2. Particle swarm optimization and portfolio constraints

Particle swarm optimization is a population based approach similar to genetic algorithm where a collection of particles move in the solution space and at each step the algorithm evaluates the fitness function at every particle moving in the space. The algorithm then calculates the new velocity for each particle to move and re-evaluate. Each particle is attracted to the best solution it found and the best location of any other particles found so far. The algorithm starts by creating initial particles from a uniform random distribution within the bounds set with an assigned position $p_i(t)$ and assigning each particle an initial velocity $v_i(t)$. After evaluating each particle's fitness function it assigns the best function $gbest_i(t)$ value and the best location $pbest_i(t)$. We then update the velocity v_i.

$$v_i(t+1) = Wv_i(t) + y_1u_1(pbest_i(t) - p_i(t)) + y_2u_2(gbest(t) - p_i(t)) \tag{4}$$

$$p_i(t+1) = p_i(t) + v_i(t+1) \tag{5}$$

where W is the inertia, y_1 and y_2 are constants, u_1 and u_2 are two uniform random numbers in (0,1), $pbest_i(t)$ is the particle i's best position so far, $p_i(t)$ is the current particle position and $gbest(t)$ is the best position in the neighborhood. PSO is a method that does not require discretization of the decision variables, it gives a very good performance with an efficient computational cost (Briza & Naval, 2011).

Based on the current literature, we apply four particle swarm optimization methods for asset selection and weights decision, and then we combine the results of each portfolio optimized under the four different PSO algorithms with the RRL long/short signals as a complete portfolio rebalancing strategy. The first PSO method is the standard particle swarm algorithm that has been used by Cura (2009) for portfolio constraint optimization. Second, we choose the improved particle swarm optimization (IPSO) proposed by Xu, Chen, and Yang (2007). Third, we apply the drift particle swarm optimization (DPSO) (Sun et al., 2011). We then apply the many optimization liaisons particle swarm optimization (MOLPSO) (Pedersen, 2010). These PSO methods are selected for the fact that each one represents a different variant to the standard PSO algorithm providing a different approach of finding a solution. In all of these methods, the fitness function is the Calmar ratio with expected maximum drawdown. We apply PSO and RRL for each asset individually and then use constraint handling techniques to construct a portfolio of assets. All the particle swarm optimization methods are applied in a multistage fashion for a dynamic asset allocation outcome. At every time step, we update the asset selection and weights dynamically.

We aim to solve some of the real world constraints faced with the portfolio optimization problems. Next, we describe the constraint handling techniques individually in the following subsections.

3.2.1. Cardinality constraint

The cardinality constraint is a restriction to the number of assets selected in a portfolio.

$$\sum_{i=1}^{N} z_i = K \tag{6}$$

where z_i is a binary value $[0, 1], i = 1, .., N,$ and N is the total number of assets to select from, and K is the number of assets selected. Here we select $K = 10$ based on a previous empirical study which concludes that a value of $K = 10$ will be sufficient and more cost efficient Liagkouras and Metaxiotis (2016). We handle this constraint by comparing the optimized fitness function value for each asset at every time step t and selecting the best K assets with the highest fitness value dynamically. A value of $z_i = 1$ will be assigned to the 10 assets with the highest fitness function values at time t. The optimization process will be repeated with a moving window where the new addition of data will change the asset selection and a value of $z_i = 1$ will be assigned to 10 new assets and a value of $z_i = 0$ is assigned to the non-selected assets. Following this method of handling the cardinality constraint will allow us to be able to satisfy the constraint at every time step t and under every PSO method in the comparison. PSO algorithms were selected in this context for the fact that they are able to find the optimal fitness values through looking in wide search area and without the need of objective function reformulation. This constraint handling technique is based on fitness function value comparison of the assets. Different fitness functions my result in different asset selection based on the criteria of the fitness function that the algorithm is trying to minimize or maximize.

3.2.2. Quantity constraint and round-lot constraint

For handling the upper and lower bound/quantity, and the round-lot constraints, we use some of the mechanisms proposed by Liagkouras and Metaxiotis (2016). We apply a lower and upper bound of $a_i = 0.01$ and $b_i = 0.99$ for each asset in the particle swarm optimization methods. The upper and lower bounds will be bound the solution during the optimization process. The quantity constraint will have an effect on the selection due to the fact that a small weight on the asset will indicate a smaller fitness function and result in the asset not being selected. Then we use a repairing technique of normalizing the selected assets. For the round-lot constraint, we round the second part of the equation to the second nearest decimal point.

$$w'_{it} = a_i z_{it} + round\left(\frac{w_{it} z_{it}}{\sum_{i=1}^{N} w_{it} z_{it}}\left(b_i z_{it} - \sum_{i=1}^{N} a_i z_{it}\right)\right) \tag{7}$$

where w_{it} is the weight of the asset i at time t generated from the PSO algorithm only with a lower and upper bound of a_i and b_i, and w'_{it} is the repaired weight after normalization and rounding. z_{it} is the binary value from the cardinality constraint solution at time t. When $z_{it} = 0$ a value of 0 will be assigned to w'_{it} Where $w'_{it} >= 0$. The lot size in this study is in the multiples of 10 stocks. We are able to solve the round lot constraint with simply rounding in Eq. (7) due to the fact that our system uses Eq. (3) to calculate the return where μ is the maximum number of shares that can be selected for an asset. In this constraint the selection of the fitness function will also have an effect on the weight selected during the optimization process. The criteria of the fitness function will decide if the asset holds the required specification that can maximize or minimize the measurement of the fitness function or not.

3.3. Data collection

In this study, we construct a portfolio with the cardinality constraint of $K = 10$. At every time step, we allocate ten assets out of the S&P100 index constituent stocks.

We extract the weekly closing prices for each asset of the S&P100 using Bloomberg. The range of data used is from January 01, 2011 to December 31, 2015. Out of the five year data, we use three years of weekly returns for training and two years for testing. While the range of data selection might affect the performance based on the market regime changes a trade-off between selecting a large data set that includes different regimes and the most recent data that might more effective exists. Since we are applying a multiperiod portfolio optimization the window of the training window will move after every decision to include the latest market data.

4. Trading algorithms comparison

We start by evaluating the best fitness function to use for the PSO methods using our training data only. We first apply the stan-

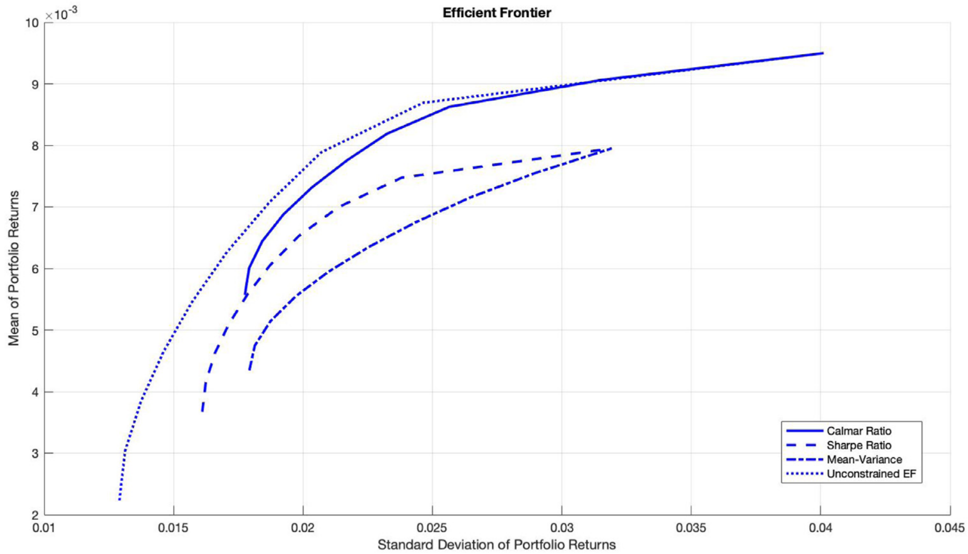

Fig. 1. Efficient frontier of PSO with different fitness functions.

dard PSO algorithm with different fitness functions. And then we apply a static optimization under the same constraints using three different fitness functions i.e. Mean-Variance, Sharpe ratio,[2] Calmar ratio. Sharpe ratio and Mean-Variance are implemented in the standard PSO as defined in Zhu et al. (2011). In Fig. 1 we generate an efficient frontier for each fitness function under the cardinality, quantity and round lot constraints and compare it with the unconstrained standard efficient frontier of the S&P100 stocks. When solving the constraints for each fitness function there will be a difference in the asset selection and the efficient frontier is plotted for the K selected assets under each fitness function. Fig. 1 shows the efficient frontier when applying the different fitness functions where the Mean-Variance fitness function includes a risk averse factor $\lambda = 0.5$ that differentiates the result from Sharpe ratio. Different values of λ can generate different asset selections (Zhu et al., 2011) where a value of $\lambda = 0$ will maximize the mean return without taking into account the variance and a value of $\lambda = 1$ will achieve the minimum variance. The frontiers are calculated based on the mean of return and standard deviation during the training period of the selected assets by each method. Since the asset selection is based on the fitness function, the main difference between the PSO with different fitness functions relies on the solution of the cardinality constraint. The figure shows that the PSO with Calmar ratio can generate optimal portfolios with higher returns with a given standard deviation. The explanation of the expected maximum drawdown and it's comparison to maximum drawdown and the relation with Sharpe ratio is done by Magdon-Ismail and Atiya (2004). While PSO with Sharpe ratio can generate optimal portfolios with a lower standard deviation. The PSO with Mean-Variance meets with Sharpe ratio at its maximum return and standard deviation portfolios. Based on the efficient frontier analysis, we select the Calmar ratio to be the fitness function for all the PSO methods, as this fitness function would provide a superior portfolio return in general.

Next, we compare the performance of four different PSO approaches combined with RRL signals. This will result in four trading algorithms producing different dynamic asset allocation decisions for the same set of assets, and then we can readily assess the merits of each combination in generating trading decisions. The resulting portfolio rebalancing methods are: The standard particle swarm optimization with RRL (RRL-PSO), the improved particle swarm optimization with RRL (RRL-IPSO), the particle swarm optimization with a drift with RRL (RRL-DPSO), and the many optimization liaisons particle swarm optimization with RRL (RRL-MOLPSO).

We show the comparison between the portfolios formulated using recurrent reinforcement learning and the different long only particle swarm optimization methods. In addition, we compare our portfolios with the S&P100 as a baseline benchmark. We use three years of weekly closing prices (January 01, 2011–January 14, 2014) to train our θ for each asset, and two years of weekly closing prices (January 15, 2014 - December 31, 2015) for testing. The total number of periods is set to 102. In order to generate the signals from the recurrent reinforcement learning model, we set the moving window of the online learning for the RRL to three years that moves forward by two weeks at every decision making step. We perform the four PSO methods with a population size of 20 and a maximum iteration of 400 for a three years training that moves forward by two weeks at every decision making step until the end of the testing period. In this test, in Eq. (3) we choose $\mu = 1000$ and $\delta = 0bp$ which is zero basis points per stock traded.

4.1. Standard particle swarm optimization with recurrent reinforcement learning portfolio (RRL-PSO)

In this method, we will apply the RRL long/short signals to the constrained portfolio weights and asset selections found using the standard PSO algorithm. The PSO simply applies Eq. (4) for the velocity calculation where $y1$ and $y2$ were set to 2 and $W = 0.9$ as recommended by many studies done on PSO methods (Poli, Kennedy, & Blackwell, 2007). Eq. (5) updates the position of each particle. Fig. 2 shows the comparison between a long only PSO constrained portfolio and a long/short (RRL-PSO) constrained portfolio against the S&P100 as the benchmark. In PSO the parameters are fixed and the selection of the parameters is critical as different parameter selection yields different performances.

4.2. Improved particle swarm optimization with recurrent reinforcement learning portfolio (RRL-IPSO)

In this model, we apply the IPSO method proposed by Xu et al. (2007). The fitness function is the Calmar ratio.

[2] Sharpe ratio in this study does not take into account any risk free rate.

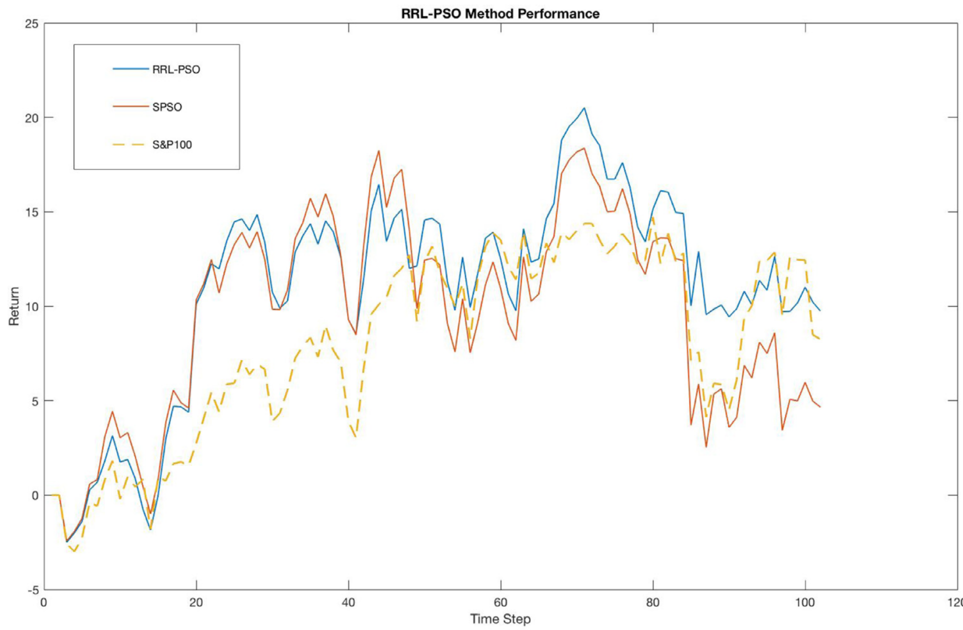

Fig. 2. Performance of RRL-PSO (L/S) portfolio and PSO long only portfolio (102 weeks).

In IPSO the parameters in Eq. (4) are updated dynamically.

$$W = 0.81 - \frac{t}{t_{max}} * 0.4$$
$$u_1 = 1 - \frac{t}{t_{max}}$$
$$u_2 = 1 + \frac{t}{t_{max}}$$

where t is the iteration and $t_{\max}$ is the maximum iteration. The velocity for the best and worst particles is calculated following Eq. (4) independently, while the position for the best particles is updated based on Eq. (5) and the worst particles position is updated using Eq. (8) such that the worst and best particles will fly oppositely.

$$p_i(t+1) = p_i(t) - v_i(t+1) \tag{8}$$

In the RRL-IPSO we make the dynamic asset selection and weight decision based on IPSO. The method will optimize its fitness function to find the weight, and the asset selection will be based on sorting the fitness function value for each asset. The upper and lower bound constraints are taken into consideration within the IPSO algorithm.

We use the signals generated by applying Eq. (2) in the RRL for each asset over the training period, and then we combine the asset selection and their weights generated from the IPSO algorithm. Let K be the number of assets in the cardinality constraint, and the weight w for each asset. This results in:

$$\sum_{i}^{K} |F_{it} w'_{it}| = 0.99 \tag{9}$$

where i is the number of assets at time t.

In the combination of IPSO and RRL, F_t in Eq. (9) is the signal from Eq. (2) and w_t is the weight from Eq. (7). Fig. 3 shows the result of the RRL-IPSO algorithm which will generate a long/short portfolio and S&P100 index as the benchmark. We also add the IPSO long only portfolio to the comparison.

4.3. Drift particle swarm optimization with recurrent reinforcement learning portfolio (RRL-DPSO)

Particle swarm optimization with a drift is the method applied in Sun et al. (2011). In this method, the particle directional movement is inspired from the drift motion of an electron in a conductor under electric field. The velocity in the DPSO is calculated following Eq. (10).

$$v_i(t+1) = \alpha \cdot |mb(t) - p_i(t)| \cdot \varphi_i(t) + y_1 u_1 (pbest_i(t) - p_i(t)) + y_2 u_2 (gbest(t) - p_i(t)) \tag{10}$$

where α is the compression-expansion coefficient, $\varphi_i(t)$ is a random number following a normal random distribution and $mb(t)$ is the mean of the *pbest* of all the particles. We set $\alpha = 0.9$ in the optimization process. Fig. 4 shows the comparison between the RRL-DPSO long/short portfolio and the DPSO long only portfolio with the S&P100 as the benchmark.

4.4. Many optimization liaisons particle swarm optimization with recurrent reinforcement learning portfolio (RRL-MOLPSO)

As in the previous experiment, the training set is three years of weekly closing prices and the testing set is two years of weekly closing prices. The PSO method applied with the recurrent reinforcement learning is the many optimizing liaisons particle swarm optimization (MOLPSO). It is a modified PSO based on the suggestions in Mezura-Montes and Coello (2011) and Pedersen (2010). The MOLPSO parameters are selected based on Pedersen (2010) shown in Table (1) and the equality and inequality constraints in the optimization process are handled based on Mezura-Montes and Coello (2011). MOLPSO is applied to the asset selection and the weights decision based on the Calmar ratio as the fitness function. In all four PSO methods, i.e. IPSO, DPSO and MOLPSO, we set the population size to 20 particles and the maximum iteration to 400 for each asset.

In Fig. 5, we show the performance of the portfolio developed using MOLPSO with the RRL (RRL-MOLPSO) that generates a

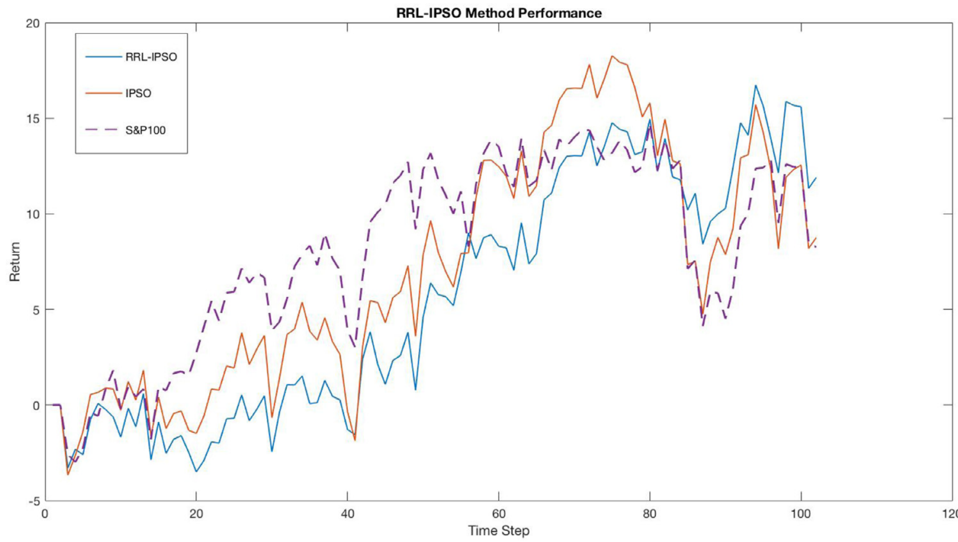

Fig. 3. Performance of RRL-IPSO (L/S) portfolio and IPSO long only portfolio (102 weeks).

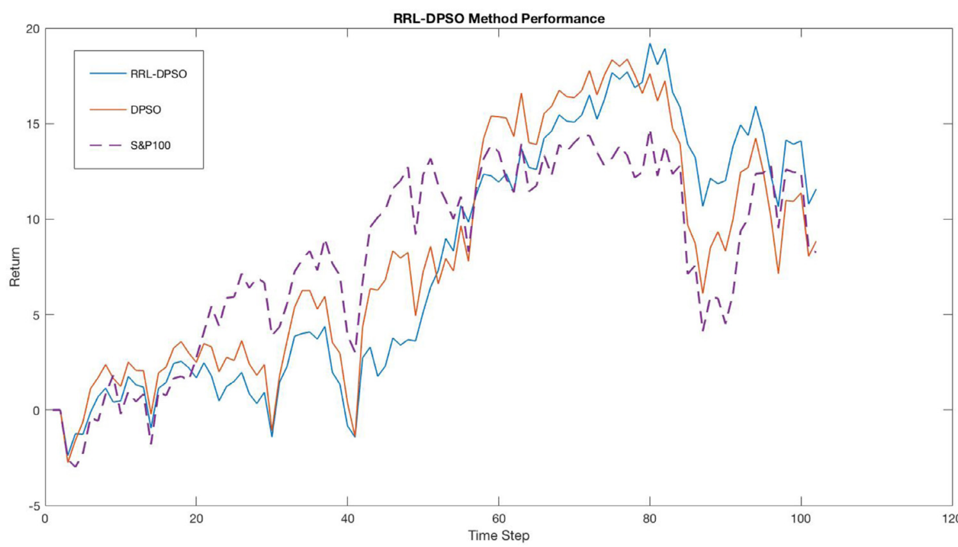

Fig. 4. Portfolio performance RRL-DPSO (L/S) portfolio and DPSO long only portfolio (102 weeks).

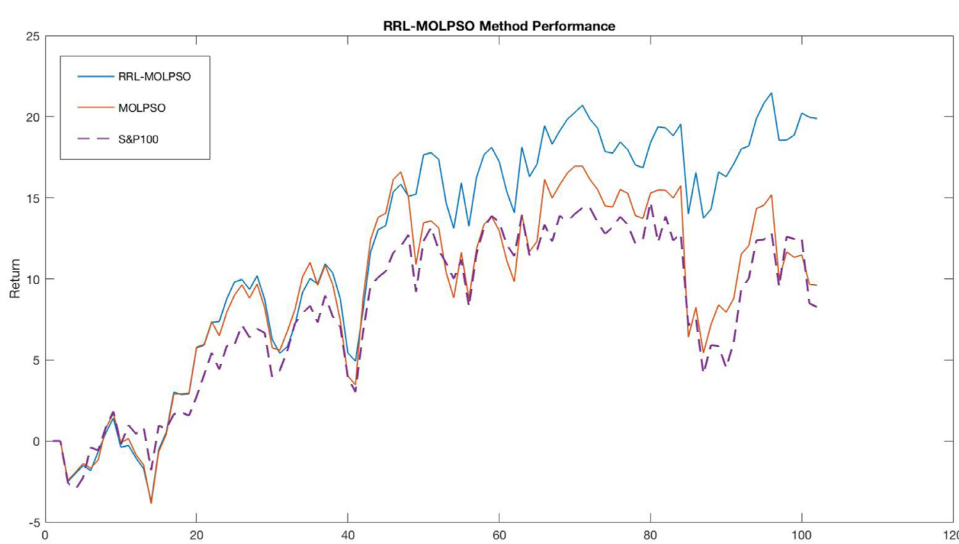

Fig. 5. Portfolio performance RRL-MOLPSO (L/S) portfolio and MOLPSO long only portfolio (102 weeks).

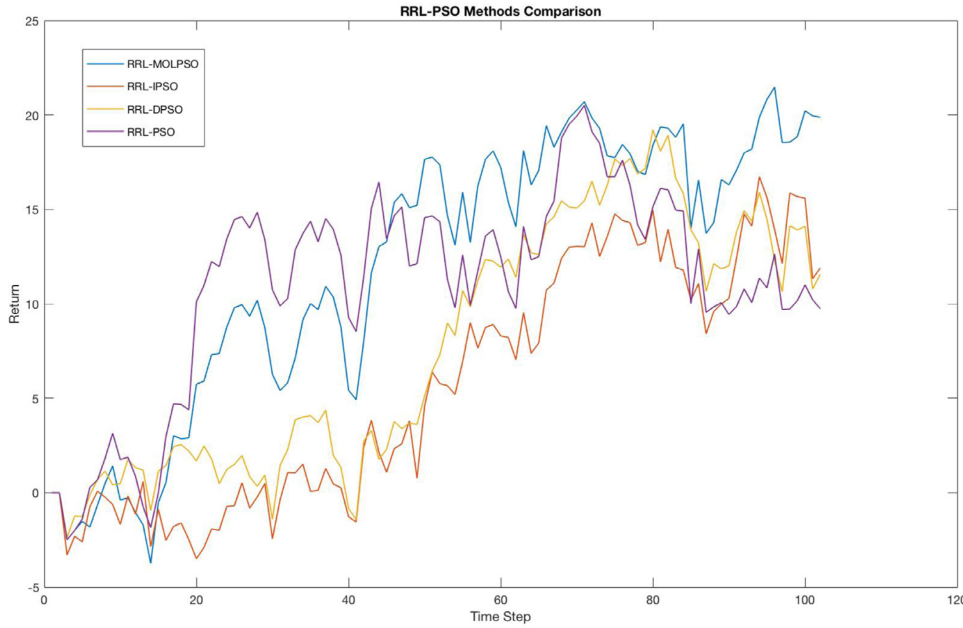

Fig. 6. Comparison in performance $\mu = 1000, \delta = 0bp$ (102 weeeks).

Table 1
MOLPSO parameters.

Problem dimensions	Fitness evaluations	Swarm size	Inertia (W)	Constant (y_2)
2	400	23	−0.3328	2.8446
		50	0.2840	1.9466
2	4000	183	−0.2797	3.0539
		139	0.6372	1.0949
5	1000	50	−0.3085	2.0273
5	10,000	96	−0.3675	4.171
10	2000	60	−0.2700	2.9708
10	20,000	116	−0.3518	3.8304
20	40,000	228	−0.3747	4.2373
20	400,000	125	−0.2575	4.6713
		67	−0.4882	2.7923
30	600,000	134	−0.4300	3.0469
50	100,000	290	−0.3067	3.6223
100	200,000	219	−0.1685	3.9162

long/short portfolio. Here the RRL-MOLPSO portfolio is compared with the MOLPSO long only portfolio and the S&P100 index. The RRL-MOLPSO portfolio is superior to the other portfolios (Fig. 6).

Table (2)[3] shows the performance of the RRL-PSO portfolio which is a long/short constrained portfolios and the PSO constrained long only portfolios with the same constraints. We observe that adding the RRL long/short signals increases the return in all the cases, but it inflects a minor decrease in the Sharpe ratio due to the increased standard deviation in returns when applying long and short trading opposed to a long only. However, when we compare Sharpe ratio with Sterling ratio,[4] we find the RRL long/short signals perform consistently better than the PSO constrained long only portfolios. This result can be explained by the high maximum drawndown generated from the PSO long only portfolios. Since the drawdown risk is generally a better measure for portfolio risks, we consider Sterling ratio as a better measure than Sharpe ratio in this case. Therefore, we conclude that the RRL long/short constrained portfolios consistently outperforms the PSO constrained long only portfolios. To the best of our knowledge the PSO methods have not been used for a constrained long/short portfolio before in the current literature.

5. Trading system and discussion

Furthermore, we develop a trading system that can handle the real world constraints and address the issue of high sensitivity to transaction costs (Fig. 7). The asset selection and weights are optimized dynamically using PSO methods while taking into account the constraints. Long/short signals are generated using Calmar ratio as both the objective function for the RRL method and the fitness function for the PSO methods. In order to handle the transaction cost, we hold the weights and asset selection changes while the RRL continues to generate long/short signals for the selected assets until the stop-loss is triggered. The MOLPSO method will re-optimize the assets and weight selections only when the stop-loss threshold is triggered clearing the unwanted assets and selecting new ones from the S&P100 with new weights.

The RRL system is trained using an online training method where the stop-loss is not applied. PSO methods are applied at time $t = 0$ to optimize the weights and asset selection and then it will only be re-optimized when the stop-loss is triggered using the data up to time $t = trigger$ which is the time that the threshold of the stop-loss condition is satisfied.

5.1. Dynamic stop-loss strategy

Given the dynamic nature of the market, a trained model needs to be retrained when the market undergoes a regime change, and the stop-loss mechanism has been proven effective in improving system performance (Lo & Remorov, 2017). The stop-loss strategy used in the proposed trading system for weight and asset selections is a simple dynamic stop-loss strategy. The notion of the

[3] The maximum drawdown calculated in the tables is the arithmetic maximum drawdown.

[4] Sterling ratio is a risk adjusted return measure similar to Sharpe ratio except that it focuses on downside risk. It measures return over average drawdown. Both Calmar ratio and Sterling ratio take the drawdown as the risk measure.

Table 2
Different RRL-PSO long/short and PSO long only constrained portfolios.

PSO portfolio ($\delta = 0$ bps)	Return (%) accumulative (annualized)	Maximum drawdown	Sharpe ratio	Sterling ratio
RRL-PSO (L/S)	9.75 (4.76)	11.08	1.9972	0.8880
PSO (long only)	4.66 (2.3)	15.84	1.7722	0.2961
RRL-IPSO (L/S)	11.99 (5.78)	6.54	0.8959	1.8480
IPSO (long only)	8.75 (4.29)	13.53	1.1615	0.6519
RRL-DPSO (L/S)	11.57 (5.63)	8.57	1.1870	1.3663
DPSO (long only)	8.84 (4.33)	12.28	1.3600	0.7260
RRL-MOLPSO (L/S)	19.89 (9.49)	6.97	1.6481	2.8943
MOLPSO (long only)	9.60 (4.69)	11.53	1.6660	0.8405
Buy & Hold	8.26 (4.13)	10.60	1.6334	0.7864

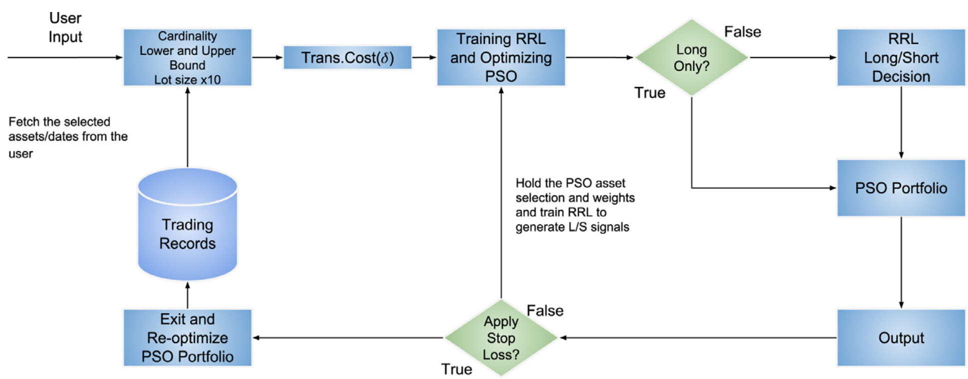

Fig. 7. Constrained portfolio trading decision system.

simple dynamic stop-loss is introduced by Chevallier, Ding, and Ielpo (2012) where it is applied to a long-only portfolio. Here we apply the concept in our trading system using the cumulative return in Eq. (11):

$$\frac{r_{t-1}}{\sigma_{t-1}} \leq -n \tag{11}$$

where r_{t-1} is the cumulative return of the portfolio up to time $t-1$, σ_{t-1} is the moving volatility up to time $t-1$, and n is the number of volatility days prompting stop-loss. The stop-loss is applied only during the testing phase and online dynamically. In Chevallier et al. (2012) the suggested value of n in Eq. (11) for a long only portfolio should be close to $n = 2$.

Fig. 8 shows the long/short signals with weights of the selected assets where the different colors of the bars indicate different assets within the trading horizon. In the background, we show the cumulative return of the S&P100. As seen in the figure, the stop loss allowed for some assets to be held in the portfolio for a longer period before switching and the RRL signals allowed the system to adapt with the market changes along with the MOLPSO asset selection under the real world constraints. The MOLPSO selects ten stocks out of the S&P100 whenever the stop-loss is triggered. The direction of the trade (long or short) of each stock is determined by the RRL algorithm. In Fig. 8, the long/short signal is shown as the direction (positive/negative) of the bar, and the weight is shown in the Y-axis of the figure in percentage, colors indicate different stocks for different time periods. The addition of stop-loss to the system limits the clearing and asset re-selection during the trading time horizon. The system will hold the weights and assets selected by the MOLPSO while the RRL will continue generating long/short signals for the assets selected until the stop-loss is triggered then MOLPSO will clear some assets and select new assets with new weights.

The weighted signals in Fig. 8 are generated following Eq. (12).

$$WS = F_{it} w'_{it} \quad i = 1, \ldots, N \quad t = 1, \ldots, T \tag{12}$$

where F_{it} is the long/short signal generated from the RRL for asset i at time step t and w'_{it} is the weight of the asset selected satisfying the constraints using MOLPSO method.

Fig. 9 shows the weights of the portfolio by sector over the investment horizon as the portfolio weights are changing dynamically. The figure demonstrates that after some time period in the beginning, the asset weight selection is focused on three sectors (i.e. health care, consumer discretionary, and industrial) primarily. Given the S&P 100 Index comprises 100 major blue chip companies across multiple industry sectors. These companies would be representative of the industry sector performances, and by grouping weights by sector we are able to gain a clear schematic view on portfolio section of the proposed method. Overall, it shows that the proposed method is able to search a broad range of sectors and eventually converges on the optimal selections in terms of industry sectors.

5.2. Transaction cost sensitivity analysis

When designing a trading system, one of the important issues is the transaction cost when executing a trading strategy (Madhavan, 2002; Tetlock, Saar-Tsechansky, & Macskassy, 2008). The transaction cost can be considered as a constraint on the return of the trading algorithm and needs to be dealt with in order to make the trading system more realistic. We developed a dynamic system that can change positions in terms of asset allocation and long/short positions to cope with the market changes. Dynamic asset allocation may result in some assets being cleared for a new asset allocation. Frequent asset clearing incurs a high transaction cost effect. In practice, the transaction costs vary depending on

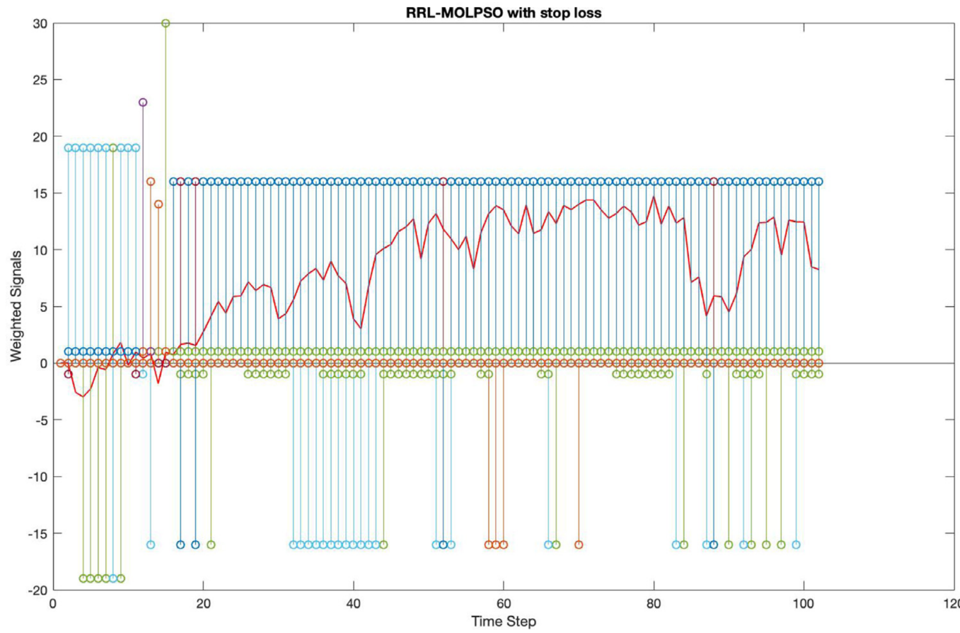

Fig. 8. Constrained portfolio trading signals.

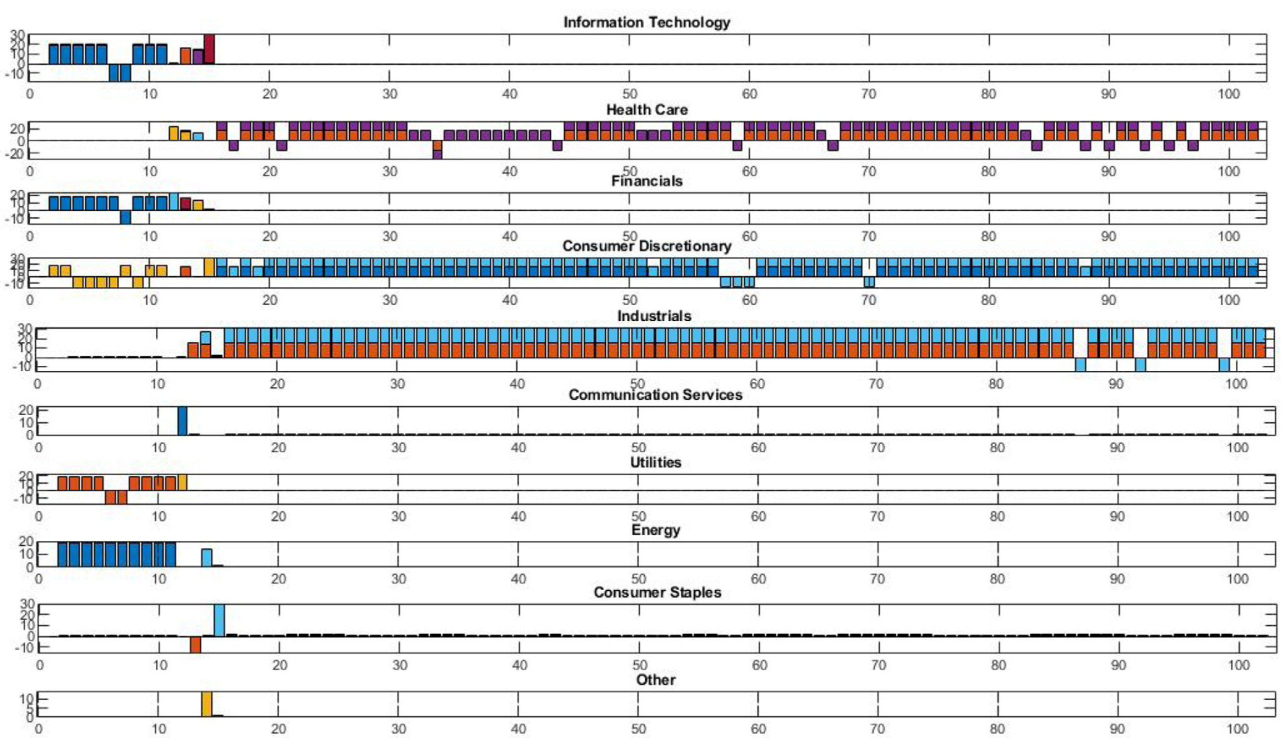

Fig. 9. Constrained portfolio trading weights by sector.

different asset classes. Based on the empirical evidence that the average round-trip trading cost of large-cap stocks on NYSE is at least 20 bps (Chan & Lakonishok, 1997; Keim & Madhavan, 1998; Mittermayer, 2004), we experiment with a range of fixed costs to their impact on the system performance to examine the sensitivity of the proposed trading system against different transaction cost levels. Trading costs (single-trip costs) of 10, 15, 20 and 25 bps are applied to the trading system, and we also examine how the addition of a stop-loss can affect the robustness of the trading system performances by limiting the dynamic switching between different assets. In this analysis a value of $n = 2$ is selected for the long only portfolio and a value of $n = 4$ is selected for the long/short portfolio as the long/short portfolio can generate a higher standard deviation due to the frequent position changes. The value of $n = 4$ for the long/short portfolio is based on trying different values $n = 2, 3, 4$ on the training data. The stop-loss threshold selection can be different under different asset selections and different transaction costs.

Portfolio ($\delta = 10$ bps)	Return (%) accu. (ann.)	Maximum drawdown	Sharpe ratio	Sterling ratio	Num. of trades
S-L (RRL-MOLPSO)	33.44 (15.52)	14.52	1.2257	2.3186	170
No S-L (RRL-MOLPSO)	1.33 (0.66)	10.76	0.8298	0.1249	1460
S-L (MOLPSO)	19.64 (9.38)	10.13	0.5999	1.9574	229
No S-L (MOLPSO)	−8.73 (−4.46)	18.51	0.0406	−0.4740	1457
S-L (RRL-IPSO)	−2.62 (−1.32)	9.17	−0.3212	−0.2893	186
No S-L (RRL-IPSO)	−1.48 (−0.74)	8.04	−0.3441	−0.1862	1415
S-L (IPSO)	13.93 (6.74)	15.17	1.5307	0.9243	29
No S-L (IPSO)	−3.97 (−2.01)	15.11	0.1811	−0.2645	1409
Portfolio ($\delta = 15$ bps)	Return (%) accu. (ann.)	Maximum drawdown	Sharpe ratio	Sterling ratio	Num. of trades
S-L (RRL-MOLPSO)	31.92 (14.86)	14.70	1.1893	2.1867	170
No S-L (RRL-MOLPSO)	−7.9465 (−4.06)	14.18	−0.5126	−0.5645	1460
S-L (MOLPSO)	−2.28 (−1.14)	12.68	−0.1766	−0.1812	503
No S-L (MOLPSO)	−17.89 (−9.39)	22.21	−0.7767	−0.8091	1457
S-L (RRL-IPSO)	−2.62 (−1.32)	12.62	−1.2130	−0.2133	18
No S-L (RRL-IPSO)	−8.17 (−4.17)	9.33	−1.8670	−0.8852	1415
S-L (IPSO)	13.78 (6.67)	15.17	1.5152	0.9148	29
No S-L (IPSO)	−10.33 (−5.31)	15.91	−0.7095	−0.6539	1409

Table 3
RRL-MOLPSO and MOLPSO long only portfolio comparison with/without stop-loss (S-L) and different transaction costs (δ). Abbreviation “accu.” stands for “accumulative”, and “ann.” stands for “annualized”.

Portfolio ($\delta = 20$ bps)	Return (%) accu. (ann.)	Maximum drawdown	Sharpe ratio	Sterling ratio	Num. of trades
S-L(RRL-MOLPSO)	30.42 (14.20)	14.88	1.1512	2.0580	170
No S-L(RRL-MOLPSO)	−17.2235 (−9.02)	18.82	−1.2081	−0.9203	1460
S-L (MOLPSO)	−30.75 (−16.78)	30.55	−1.4686	−1.0099	1427
No S-L (MOLPSO)	−27.06 (−14.59)	27.73	−1.1469	−0.9794	1457
S-L (RRL-IPSO)	−16.55 (−8.65)	19.32	−1.4615	−0.8615	813
No S-L (RRL-IPSO)	−14.85 (−7.73)	14.83	−2.5004	−1.0083	1415
S-L (IPSO)	13.64 (6.60)	15.17	1.4996	0.9053	29
No S-L (IPSO)	−16.69 (−8.73)	17.01	−1.3335	−0.9878	1409
Portfolio ($\delta = 25$ bps)	Return (%) accu. (ann.)	Maximum drawdown	Sharpe ratio	Sterling ratio	Num. of trades
S-L(RRL-MOLPSO)	28.91 (13.54)	15.06	1.1111	1.9324	170
No S-L(RRL-MOLPSO)	−26.50 (−14.27)	25.51	−1.4400	−1.0427	1460
S-L (MOLPSO)	−38.5838 (−21.63)	38.33	−1.5388	−1.0092	1437
No S-L (MOLPSO)	−36.224 (−20.14)	34.65	−1.3236	−1.0483	1457
S-L (RRL-IPSO)	−20.82 (−11.02)	23.46	−1.3882	−0.8913	872
No S-L (RRL-IPSO)	−21.54 (−11.42)	21.37	−2.4011	−1.0126	1415
S-L (IPSO)	13.49 (6.54)	15.17	1.4839	0.8958	29
No S-L (IPSO)	−23.06 (−12.28)	22.95	−1.5972	−1.0091	1409

In Table 3, we show the performances of the RRL-MOLPSO long/short constrained portfolio and the MOLPSO long only constrained portfolio with stop-loss (S-L) and without stop-loss (No S-L) controls at different transactions cost levels. We see that the stop-loss is able to make the portfolio endure higher transaction costs in the case that $\delta \geq 10$ bps as the stop-loss strategy will exit the market when the volatility is high, and then and only then it will initiate the MOLPSO optimization in order to change the weight and asset selection of the portfolio. Moreover, the RRL-MOLPSO outperforms all other portfolios at all cost levels when they are measured by Sterling ratio and annualized accumulative return. The IPSO long only portfolio outperforms in Sharpe ratio. The RRL long/short signals effect is minimized or maximized by the weight of the asset which changes in position.

6. Conclusion

In this study, we extend the unconstrained recurrent reinforcement learning portfolio trading system to a constrained portfolio optimization and trading system. We add several common portfolio optimization constraints, i.e. the cardinality constraint, floor and ceiling and round lot constraint, by applying the MOLPSO method using the Calmar ratio with an expected maximum drawdown as the fitness function. Based on an efficient frontier and a transaction cost analysis, we conclude that the Calmar ratio is the best fitness function to be used for the PSO algorithms. Moreover, we develop a portfolio trading system where the recurrent reinforcement learning algorithm is trained using an online training method to generate the long/short signals dynamically, while other studies apply the technical trading rules with weights and the long/short signals are generated by PSO optimization methods (Briza & Naval, 2011; Wang, Philip, & Cheung, 2014; Worasucheep, Nuannimnoi, Khamvichit, & Attagonwantana, 2017). We apply a set of selected PSO methods in a dynamic optimization setting and then propose a stop-loss control over the re-optimization of the system. Overall, we show that the proposed active portfolio trading system with RRL-MOLPSO algorithm and stop-loss control is superior than the other benchmark portfolio trading systems with the same portfolio constraints where the stocks are dynamically selected from the S&P100 index constituent stocks.

The addition of the RRL method extends the constrained portfolio optimization to a long/short constrained portfolio optimization and trading system. In this study, we focus on applying a single meta-heuristic algorithm with RRL to solve the constrained portfolio problem. Other meta-heuristic algorithms may also be combined with RRL to create a constrained long/short portfolio. A comparison between the different meta-heuristic methods with RRL can be done for future studies.

The constrained portfolio in this paper had a cardinality, floor and ceiling, and a round lot constraints on equity only portfolios. The addition of multiple asset classes might be added with asset class constraints. Adding the asset class constraints can introduce the complexity of the problem under transaction costs since every class might have a different transaction cost calculation. It may also add to the complexity of the other constraints and the objective functions.

While we apply RRL for the long/short decision in the proposed trading system, other methods such as neural networks, genetic programming, and support vector machine methods may also be applied for the same purpose (Chiang, Enke, Wu, & Wang, 2016). For future studies, one may explore the effect of deep learning methods such as the deep recurrent reinforcement learning on the constrained portfolio optimization where an improved recurrent network may help better capture price movement patterns.

Credit authorship contribution statement

Saud Almahdi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing - original draft. **Steve Y. Yang:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing - review & editing.

Acknowledgement

The authors also acknowledge that Saud Almahdi has been funded by the Saudi Arabian Cultural Mission (SACM) fellowship during his Doctoral study at Stevens Institute of Technology, and Steve Y. Yang was an Assistant Professor in the School of Business at Stevens Institute of Technology. There are no other affiliations with or involvement in other organizations or entities with financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications, 87*, 267–279.
Briza, A. C., & Naval, P. C. (2011). Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data. *Applied Soft Computing, 11*(1), 1191–1201.
Chan, L. K. C., & Lakonishok, J. (1997). Institutional equity trading costs: NYSE versus Nasdaq. *The Journal of Finance, 52*(2), 713–735.
Chang, T.-J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research, 27*(13), 1271–1302.
Chen, Y., Mabu, S., Hirasawa, K., & Hu, J. (2007). Genetic network programming with sarsa learning and its application to creating stock trading rules. In *Evolutionary computation, 2007. CEC 2007. IEEE congress on* (pp. 220–227). IEEE.
Chen, Y., Mabu, S., Shimada, K., & Hirasawa, K. (2009). A genetic network programming with learning approach for enhanced stock trading model. *Expert Systems with Applications, 36*(10), 12537–12546.
Chevallier, J., Ding, W., & Ielpo, F. (2012). Implementing a simple rule for dynamic stop-loss strategies. *The Journal of Investing, 21*(4), 111–114.
Chiang, W.-C., Enke, D., Wu, T., & Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications, 59*, 195–207.
Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications, 10*(4), 2396–2406.
Deng, G.-F., Lin, W.-T., & Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications, 39*(4), 4558–4566.
Gabrielsson, P., & Johansson, U. (2015). High-frequency equity index futures trading using recurrent reinforcement learning with candlesticks. In *Computational intelligence, 2015 IEEE symposium series on* (pp. 734–741). IEEE.
Golmakani, H. R., & Fazel, M. (2011). Constrained portfolio selection using particle swarm optimization. *Expert Systems with Applications, 38*(7), 8327–8335.
Gorse, D. (2010). Application of stochastic recurrent reinforcement learning to index trading. *ESANN 2011 proceedings, 19th European symposium on artificial neural networks, computational intelligence and machine learning*, 123–128.
Kalayci, C. B., Ertenlice, O., Akyer, H., & Aygoren, H. (2017). An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Systems with Applications, 85*, 61–75.
Keim, D. B., & Madhavan, A. (1998). The cost of institutional equity trades. *Financial Analysts Journal, 54*(4), 50–69.
Kennedy, R. (1995). J. and eberhart, particle swarm optimization. In *Proceedings of IEEE international conference on neural networks IV, pages: 1000*.
Li, J., & Chan, L. (2006). Reward adjustment reinforcement learning for risk-averse asset allocation. In *Neural networks, 2006. IJCNN'06. International joint conference on* (pp. 534–541). IEEE.
Liagkouras, K., & Metaxiotis, K. (2016). A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem. *Annals of Operations Research*, 1–39.
Lo, A. W., & Remorov, A. (2017). Stop-loss strategies with serial correlation, regime switching, and transaction costs. *Journal of Financial Markets, 34*, 1–15.
Madhavan, A. N. (2002). Implementation of hedge fund strategies. *Special Issues (Hedge Fund Strategies: A Global Outlook), 2002*(1), 74–80.
Magdon-Ismail, M., & Atiya, A. F. (2004). Maximum drawdown. *Risk Magazine, 17*(10), 99–102.
Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91.
Mezura-Montes, E., & Coello, C. A. C. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation, 1*(4), 173–194.
Mittermayer, M.-A. (2004). Forecasting intraday stock price trends with text mining techniques. In *System sciences, 2004. Proceedings of the 37th annual hawaii international conference on* (pp. 10–pp). IEEE.
Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Science, 17*(February 1997), 441–470.
Moral-Escudero, R., Ruiz-Torrubiano, R., & Suárez, A. (2006). Selection of optimal investment portfolios with cardinality constraints. In *Evolutionary computation, 2006. CEC 2006. IEEE congress on* (pp. 2382–2388). IEEE.
Neuneier, R. (1998). Enhancing q-learning for optimal asset allocation. In *Advances in neural information processing systems* (pp. 936–942).
Ni, Q., Yin, X., Tian, K., & Zhai, Y. (2017). Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem. *Natural Computing, 16*(1), 31–44.
Pedersen, M. E. H. (2010). Good parameters for particle swarm optimization. *Tech. Rep. HL1001*. Copenhagen, Denmark: Hvass Lab.
Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence, 1*(1), 33–57.
Sun, J., Fang, W., Wu, X., Lai, C.-H., & Xu, W. (2011). Solving the multi-stage portfolio optimization problem with a novel particle swarm optimization. *Expert Systems with Applications, 38*(6), 6727–6735.
Sutton, R. S., Barto, A. G., & Williams, R. J. (1992). Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems, 12*(2), 19–22.
Tetlock, P. C., Saar-Tsechansky, M., & Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance, 63*(3), 1437–1467.
Wang, F., Philip, L., & Cheung, D. W. (2014). Combining technical trading rules using particle swarm optimization. *Expert Systems with applications, 41*(6), 3016–3026.
Worasucheep, C., Nuannimnoi, S., Khamvichit, R., & Attagonwantana, P. (2017). An automatic stock trading system using particle swarm optimization. In *Electrical engineering/electronics, computer, telecommunications and information technology (ecti-con), 2017 14th international conference on* (pp. 497–500). IEEE.
Xu, F., Chen, W., & Yang, L. (2007). Improved particle swarm optimization for realistic portfolio selection. In *Software engineering, artificial intelligence, networking, and parallel/distributed computing, 2007. SNPD 2007. Eighth ACIS international conference on: 1* (pp. 185–190). IEEE.
Zhang, J., & Maringer, D. (2013). Indicator selection for daily equity trading with recurrent reinforcement learning. In *Proceedings of the 15th annual conference companion on genetic and evolutionary computation* (pp. 1757–1758). ACM.
Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle swarm optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications, 38*(8), 10161–10169.