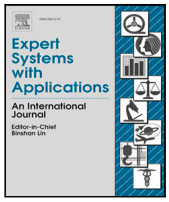
Expert Systems With Applications 254 (2024) 124299

Contents lists available at ScienceDirect

Expert Systems With Applications

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



An asset subset-constrained minimax optimization framework for online portfolio selection

Jianfei Yin [a], Anyang Zhong [a], Xiaomian Xiao [a], Ruili Wang [b,c,*], Joshua Zhexue Huang [a]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[b] School of Mathematical and Computational Sciences, Massey University, Auckland, New Zealand
[c] School of Artificial Intelligence, Dalian Maritime University, Dalian, China

ARTICLE INFO

Dataset link: https://github.com/xiterator/ASCM

Keywords:
Asset subset-constrained online portfolio selection
Minimax optimization
Robust online portfolio selection
Projected subgradient method

ABSTRACT

Effective online portfolio selection necessitates seamless integration of three key properties: diversity, sparsity, and risk control. However, existing algorithms often prioritize one property at the expense of the others due to inherent conflicts. To address this issue, we propose an asset subset-constrained minimax (ASCM) optimization framework, which generates optimal portfolios from diverse investment strategies represented as asset subsets. ASCM consists of: (i) a minimax optimization model that focuses on risk control by considering a set of loss functions constrained by different asset subsets; (ii) the construction of asset subsets via price-feature clipping, which effectively reduces redundant assets in the portfolio; (iii) a state-based estimation of price trends that guides all ASCM loss functions, facilitating the generation of sparse solutions. We solve the ASCM minimax model using an efficient iterative updating formula derived from the projected subgradient method. Furthermore, we achieve near $O(1)$ time complexity through a novel initialization scheme. Experimental results demonstrate that ASCM outperforms eight representative algorithms, including the best constant rebalanced portfolio in hindsight (BCRP) on five out of the six real-world financial datasets. Notably, ASCM achieves a 67-fold improvement over BCRP in cumulative wealth on the TSE dataset.

1. Introduction

The task of online portfolio selection presents significant challenges for machine intelligence systems, requiring trading agents with substantial financial intelligence to navigate volatile financial markets (Gunjan & Bhattacharyya, 2023; Rahwan et al., 2019; Yin, Wang, Ju, Bai, & Huang, 2020). In this environment, trading agents strategically allocate their initial capital among multiple tradable assets in each trading period, aiming to maximize cumulative wealth (CW) through investment rewards (Cover, 1991; Lai & Yang, 2022; Li & Hoi, 2014). However, these rewards are accompanied by inherent risks, as the non-stationary nature of financial markets can nullify prior earnings due to unpredictable events, resulting in capital fluctuations, reductions, or even bankruptcy (Livan, Inoue, & Scalas, 2012; Spelta, Pecora, & Pagnottoni, 2022). Furthermore, given the competition among numerous trading agents for limited capital resources, a portfolio strategy that succeeds in one market trend may fail in the next, emphasizing the need for swift adjustments to new market conditions (Sengupta, 1989; Treynor, 1999).

Given the non-stationary and zero-sum natures of financial markets, the successful integration of three crucial properties–diversity, sparsity, and risk control–becomes crucial for the effectiveness of online portfolio selection (Anis & Kwon, 2022; Bai, Yin, Ju, Chen, & Huang, 2020; Brodie, Daubechies, De Mol, Giannone, & Loris, 2009; Zheng, Hospedales, & Yang, 2020). However, a challenge arises in developing a concise and efficient optimization method that seamlessly incorporates these aspects. The following analysis provides insights into the reasons behind this challenge.

A diversified portfolio, guided by Markowitz's mean–variance theory, aims to select assets with weak or negative return correlation, providing downside protection by leveraging the potential for better returns from certain assets to offset the poor performance of others (Lai & Yang, 2022). However, implementing mean–variance optimization for online portfolio selection faces several challenges. Firstly, accurately calculating the covariance matrix in the short term becomes challenging due to the volatility and limited number of available samples for asset returns (Lai & Yang, 2022; Ledoit & Wolf, 2022; Petukhina, Klochkov, Härdle, & Zhivotovskiy, 2023). Secondly, many

* Corresponding author at: School of Mathematical and Computational Sciences, Massey University, Auckland, New Zealand.
E-mail addresses: yjf@szu.edu.cn (J. Yin), zhonganyang2021@email.szu.edu.cn (A. Zhong), xiaoxiaomian2021@email.szu.edu.cn (X. Xiao), ruili.wang@massey.ac.nz (R. Wang), zx.huang@szu.edu.cn (J.Z. Huang).

https://doi.org/10.1016/j.eswa.2024.124299
Received 15 October 2023; Received in revised form 20 March 2024; Accepted 21 May 2024
Available online 29 May 2024

mean–variance-based models heavily rely on the assumption of specific statistical distributions for asset returns, which may not be consistent with real-world cases (Chen, Wiesel, & Hero, 2011; Lai & Yang, 2022; Steinbach, 2001). Thirdly, the passive nature of downside protection provided by traditional mean–variance optimization, which requires holding the selected assets for many trading periods (Leippold, Trojani, & Vanini, 2011), can lead to suboptimal capital performance.

In the context of online portfolio selection, it is crucial for a diversified portfolio to align with short-term portfolio optimization models, particularly Kelly's capital growth theory, as highlighted by several researchers (Cover, 1991; Lai & Yang, 2022; Li & Hoi, 2014; Li, Hoi, Sahoo, & Liu, 2015; MacLean, Thorp, & Ziemba, 2010). This approach is preferred over solely relying on adapting mean–variance methodology to the online scenario (Baron, Brogaard, Hagströmer, & Kirilenko, 2019). While algorithms like short-term portfolio optimization with loss control (SPOLC) (Lai, Tan, Wu, & Fang, 2020) utilize sophisticated matrix factorization for short-term correlation matrix estimation, they may still underperform trend-tracking based approaches, such as peak price tracking (PPT) (Lai, Dai, Ren, & Huang, 2017) and Gaussian weighting reversion (GWR) (Cai & Ye, 2019), on specific datasets.

An alternative approach to achieving a diversified portfolio is through the ensemble of multiple portfolio selection algorithms, as demonstrated in recent studies (Niu, Li, & Li, 2022; Shavandi & Khedmati, 2022; Tsantekidis, Passalis, & Tefas, 2021; Yang, Liu, Zhong, & Walid, 2020). In this context, the portfolio selection algorithms themselves are treated as experts or agents. By combining the portfolio advice provided by these experts, portfolio ensemble algorithms can effectively mitigate downside risk and improve generalization capabilities (Song et al., 2023; Yang, Liu, et al., 2020). However, these methods rely on the assumption of behavior diversity exhibited by experts in an expert pool, rather than directly utilizing asset diversity represented by various asset subsets. Consequently, the performance of these methods may be suboptimal if the behavior diversity is not promised within the expert pool under certain market situations. This issue becomes even more evident since these methods lack a systematic approach to constructing an expert pool with a hedging trading structure in mind (Daluiso, Pinciroli, Trapletti, & Vittori, 2023; Tsantekidis et al., 2021). Additionally, they often employ the same formulation of reward to train all agents in the expert pool, overlooking the need for diversified training objectives (Niu et al., 2022).

A sparse portfolio aims to improve returns by concentrating capital into a select few assets rather than spreading it across the entire set of assets. Many portfolio optimization algorithms incorporate sparse constraints, such as vector norms or set cardinality, to generate sparse portfolios (Anis & Kwon, 2022; Dai & Kang, 2022; Kobayashi, Takano, & Nakata, 2021; Li, Shi, Leung, & So, 2022; Shi, Li, Leung, & So, 2022; Wei, Liu, & Fan, 2022). However, these algorithms have certain limitations. Firstly, the assets selected by a sparse portfolio may not effectively represent the diverse risk-return profiles of assets (Aquino, Sornette, & Strub, 2023). For example, asset subsets obtained through l_1-norm constraints may overlook assets identified by technical indicators, which have proven effective in predicting price movements (Ji, Yu, Hu, Xie, & Ji, 2022; Shynkevich, McGinnity, Coleman, Belatreche, & Li, 2017). Secondly, these algorithms typically focus on single-objective optimization and overlook the benefits of multi-objective optimization in managing short-term risks across different strategies (Chen & Zhou, 2022). Consequently, there is an urgent need to explore new approaches that can better align sparse portfolios with properly diversified portfolios in the context of online portfolio selection.

In addition to mean–variance optimization, various risk control mechanisms, such as conditional value-at-risk (CVaR) (Lai, Li, Wu, Guan, & Fang, 2022; Li, Qin, & Yan, 2022; Luan, Zhang, & Liu, 2022; Zheng & Zheng, 2022), and safety-first principle (Chiu, 2021; Li & Mi, 2021), have been proposed for portfolio selection. However, these methods exhibit certain drawbacks: (i) The formalism involving CVaR and safety-first principle requires the computation of probability inequalities or expectations, which is challenging to achieve statistical validity in the short term (Lai & Yang, 2022; Petukhina et al., 2023). (ii) They prioritize risk control as one of their main objectives, often overlooking other important goals such as achieving sparse portfolios, strategy aggregation (He & Yang, 2022; Yang, He, Xian, Lin, & Zhang, 2020), and strategy switching (Li, Chen, Feng, & Ying, 2017). For instance, a recent study by Lai et al. (2022) introduced the multitrend conditional value at risk (MT-CVaR) algorithm, which incorporates an approximation of CVaR as a regularization term in its objective. However, this regularization approach may lead to the generation of non-sparse portfolios in certain real-world financial markets.

Based on the analysis above, **a crucial insight for providing a concise and efficient optimization method to integrate diversity, sparsity, and risk control for online portfolio selection is to leverage asset subsets as the fundamental building blocks rather than portfolio vectors**. This approach enables the construction of a hierarchical optimal model for online portfolio selection. By utilizing various asset subsets, portfolio diversification can be achieved, and sparse portfolios are generated by optimizing a short-term trend tracking objective within the intersection of these asset subsets. Furthermore, the risk control across different investment strategies, represented by these asset subsets, can be implemented by constructing a minmax optimization on the resulting sparse portfolios.

In summary, our main contributions are:

- We propose an innovative optimization framework called asset subset-constrained minimax (ASCM)[1] to address the challenge of concise and effective online portfolio selection that integrates diversity, sparsity, and risk control. ASCM incorporates three key components: (i) A minimax optimization model that effectively controls risks by simultaneously optimizing a set of loss functions constrained by asset subsets. This approach provides a flexible and robust risk management strategy as it considers worst-case performance estimation for short-term portfolio optimization. (ii) A price-feature clipping method is employed for the construction of asset subsets. This method effectively reduces redundant assets in the portfolio. (iii) A state-based estimation of price trends based on first-order and second-order differential approximation of price dynamics. The resulting price-trend vectors are plugged into the minimax optimization model, guiding all ASCM loss functions to generate sparse portfolio vectors.
- We solve the ASCM minimax model by employing an effective iterative updating formula derived from the projected subgradient method. Through rigorous proof, we demonstrate that this updating formula achieves sub-linear convergence $O(\frac{1}{\epsilon^2})$ even under random initialization. Additionally, we introduce a greedy initial method for setting the initial feasible solution of the ASCM minimax model. This approach greatly improves the convergence speed of the updating formula, leading to an approximate time complexity of $O(1)$. The remarkable efficiency of this iterative process enhances the practicality and applicability of the ASCM minimax model, especially in high-frequency portfolio trading scenarios where time-sensitive decision-making plays a critical role.
- To evaluate the extensibility and performance of the ASCM framework, we implemented three instances of ASCM algorithms: ASCM, ASCM_{α}, and ASCM_{β}. Extensive experiments were conducted using six real-world financial datasets. The results demonstrate that the ASCM framework effectively supports diversified investment strategies, generates manageable sparse portfolios, and proficiently controls risks across multiple investment strategies. Furthermore, ASCM outperforms eight representative algorithms, including the best constant rebalanced portfolio in hindsight (BCRP), in terms of cumulative wealth across five out of

[1] The source codes are available at https://github.com/xiterator/ASCM.

the six real-world financial datasets. Remarkably, ASCM achieves a 67-fold improvement over BCRP in cumulative wealth on the TSE dataset.

The paper is structured as follows: Section 2 provides a review of related work in online portfolio selection. Section 3 outlines the problem setting, challenges, and objectives. Section 4 introduces the ASCM optimization framework, discussing its three key components. Section 5 presents the ASCM algorithm, including the initialization scheme and estimation of time complexity. Section 6 presents extensive experiments and analysis. Finally, Section 7 summarizes the key findings, contributions, and suggests potential future research directions.

2. Related work

This section provides an insightful overview of previous research on online portfolio selection, encompassing various related topics such as portfolio diversity, sparsity, and risk control.

2.1. Universal portfolio selection

The field of online portfolio selection, based on Kelly's capital growth theory (Li & Hoi, 2014; Li et al., 2015), revolves around high-frequency trading and necessitates daily position adjustments to maximize cumulative wealth. Early universal algorithms, such as universal portfolios (UP) (Cover, 1991), online Newton step (ONS) (Agarwal, Hazan, Kale, & Schapire, 2006), and exponential gradient (EG) (Helmbold, Schapire, Singer, & Warmuth, 1998), were developed with the goal of achieving universal regret bounds. These algorithms exhibit respective logarithmic regret bounds of $O(n\log T)$, $O(Gn\log T)$, and $O(G\sqrt{T\log n})$ when compared to the best constant rebalanced portfolio (BCRP) in hindsight. Here, n denotes the number of assets, T represents the number of periods, and the value of G can be unbounded (Luo, Wei, & Zheng, 2018). Recent advancements in the field of universal regret bounds have been made, as demonstrated by the works of Bhatt, Ryu, and Kim (2023), He and Yang (2022). For instance, Bhatt et al. (2023) extended the seminal framework of universal portfolios introduced by Cover and Ordentlich (1996) to accommodate continuous side information sequences. Additionally, Mhammedi and Rakhlin (2022) introduced the damped online Newton step (DONS) algorithm, which is characterized by per-round time and space complexities that scale logarithmically with the investment horizon.

While these universal algorithms can be proven to achieve universal regret bounds compared to BCRP, they may not align with the objective of Kelly's capital growth in practical scenarios. Non-universal algorithms, such as online portfolio selection with moving average reversion (OLMAR) (Li et al., 2015), short-term portfolio optimization with loss control (SPOLC) (Lai et al., 2020), trend peak price tracing (TPPT) (Dai, Liang, Dai, Huang, & Adnan, 2022), and wealth flow model (WFM) (Yin et al., 2021), have consistently demonstrated superior performance over universal algorithms, such as UP (Cover, 1991), ONS (Agarwal et al., 2006), and EG (Helmbold et al., 1998), across numerous real-world datasets. These non-universal algorithms even frequently outperform the benchmark algorithm BCRP, highlighting their efficacy in practice.

2.2. Multi-period mean–variance methods

Markowitz revolutionized portfolio diversification with his mean–variance framework, which optimizes the balance between return and risk by selecting assets with low or negative correlation. However, this framework relies on the assumption of specific return distributions and requires precise estimation of model parameters (Ledoit & Wolf, 2022; Petukhina et al., 2023), making it unsuitable for frequent rebalancing in online portfolio selection (Lai & Yang, 2022).

To address this limitation, adapting the mean–variance framework for online environments represents a viable strategy to enhance portfolio diversity and risk control (Lezmi, Roncalli, & Xu, 2022; Wu, Xie, Ge, & De Simone, 2024). For example, the short-term portfolio optimization with loss control (SPOLC) algorithm introduced by Lai et al. (2020), defined as follows:

$$\begin{aligned}&\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \gamma \boldsymbol{z}^{\intercal}\mathbf{H}\boldsymbol{z} - q\\ &\text{s.t.}\quad \mathbf{X}\boldsymbol{b} \geq q\mathbf{1},\ \boldsymbol{b}\in\Delta^{n-1},\ \boldsymbol{z}=[\boldsymbol{b};q].\end{aligned} \tag{1}$$

In this formulation, the matrix $\mathbf{H}\in\mathbb{R}^{(n+1)\times(n+1)}$ represents a rank-one covariance estimation of asset returns, while the matrix $\mathbf{X}\in\mathbb{R}^{w\times n}$ stores relative price samples within a recent time window of length w. The variable q represents a lower bound for returns over the w periods, serving as a short-term proxy for the risk tolerance factor in Markowitz's mean–variance model. It is jointly optimized alongside the portfolio vector $\boldsymbol{b}$.

Another approach proposed by Li, Uysal, and Mulvey (2022) follows the mean–variance framework and introduces a predictive control model:

$$\begin{aligned}&\min_{\{\boldsymbol{b}_\tau\}} \sum_{\tau=t+1}^{t+w} \gamma_{\text{risk}}\boldsymbol{b}_\tau^{\intercal}\hat{\boldsymbol{\Sigma}}_\tau\boldsymbol{b}_\tau + \gamma_{\text{trade}}\left\|\boldsymbol{b}_\tau - \boldsymbol{b}_{\tau-1}\right\|_1 - \hat{\boldsymbol{r}}_\tau^{\intercal}\boldsymbol{b}_\tau\\ &\quad\text{s.t.}\quad \boldsymbol{b}_\tau\in\Delta^{n-1}.\end{aligned} \tag{2}$$

In this model, the covariance matrix $\hat{\boldsymbol{\Sigma}}_\tau\in\mathbb{R}^{n\times n}$ and the return vector $\hat{\boldsymbol{r}}_\tau\in\mathbb{R}^n$ are predicted using a hidden Markov model. The objective function combines the minimization of covariance-based risk measures, transaction costs, and the predicted next-period returns. The model (2) utilizes the sum of the returns $\hat{\boldsymbol{r}}_\tau^{\intercal}\boldsymbol{b}_\tau$ over the w window as a proxy for the risk tolerance, which differs from the lower bound return q in model (1). However, both models (1) and (2) often generate non-sparse portfolio vectors, which can lead to inefficient capital utilization and suboptimal portfolio performance in certain market scenarios.

2.3. Sparse portfolio selection

To harness efficiency gains from concentrating investments in a select few assets, sparsity constraints are commonly incorporated into portfolio optimization models (Kobayashi et al., 2021; Li, Shi, et al., 2022; Li, Zhang, Wang, & Bai, 2023; Luo, Yu, Xiu, & Wang, 2022; Shi et al., 2022; Wei et al., 2022; Wu et al., 2024). For example, Wei, Yang, Jiang, and Liu (2021) proposed the dynamic sentiment-adjusted model (DSAM) to identify a subset of k assets, utilizing a stochastic neural network algorithm to construct these assets. Similarly, Shi et al. (2022) introduced a cardinality-constrained model formulated as follows:

$$\begin{aligned}&\min_{\boldsymbol{b},\boldsymbol{z}} \boldsymbol{b}^{\intercal}\boldsymbol{\Sigma}\boldsymbol{b} - \lambda\hat{\boldsymbol{r}}^{\intercal}\boldsymbol{b} + \frac{C}{2}\left(\boldsymbol{b}^{\intercal}\mathbf{1}-1\right)^2 + \mathcal{I}(\boldsymbol{z}),\\ &\qquad\text{s.t.}\quad \boldsymbol{b}=\boldsymbol{z}\end{aligned} \tag{3}$$

where the indicator function $\mathcal{I}(\boldsymbol{z})$ is defined as $\mathcal{I}(\boldsymbol{z})=0$ when the condition $\|\boldsymbol{z}\|_0\leq k$ is satisfied, and $\mathcal{I}(\boldsymbol{z})=+\infty$ otherwise. The parameter k is a preset constant, and its optimal value may vary depending on different testing datasets.

Recognizing the importance of considering portfolio diversity in the design of the optimization objective, it becomes evident that relying solely on a sparsity constraint is inadequate for achieving robust portfolio performance. In the context of online portfolio selection, if we define portfolio sparsity as the requirement of outputting a sparse portfolio vector in each trading period, many existing mean–variance based methods do not meet this standard. This underscores the inherent conflict between portfolio sparsity and diversity (Yin, Huang, Yang, & Hao, 2014). To address this conflict, this paper introduces a novel approach to portfolio diversity, represented by a collection of asset subsets, aiming to reconcile these opposing objectives.

2.4. Probability measure on risks

In addition to return-covariance-based risk measures, alternative approaches for risk management in online portfolio selection include the utilization of probability measures, such as value-at-risk (Alameer & Al Shehri, 2022; Lai et al., 2022; Leung & Wang, 2020; Li, Qin, & Yan, 2022), and the application of the safety-first principle (Chiu, 2021; Li & Mi, 2021). These approaches are grounded in measure-theory principles, providing a robust foundation for evaluating risks in portfolio selection. For example, Lai et al. (2022) introduced a portfolio optimization model with a multi-trend conditional value-at-risk (CVaR) measure as follows:

$$\begin{aligned} \min_{\boldsymbol{b},a}\ & (1-k)\left(a+\frac{1}{1-c}\sum_{l=1}^{L}\psi_l\mathbb{E}_l[-r(\mathbf{b})-a]^+\right)-k\hat{\boldsymbol{u}}^\top\boldsymbol{b} \\ \text{s.t.}\ & \boldsymbol{b}^\top\mathbf{1}=1,\ \left\|\boldsymbol{b}-\hat{\boldsymbol{b}}_t\right\|_2^2\le\epsilon,\ \epsilon>0. \end{aligned} \tag{4}$$

In this model, the expectation $\mathbb{E}_l[-r(\boldsymbol{b})-a]^+$ computes the probability $\Pr\{-r(\boldsymbol{b})-a\ge 0\}$, where $r(\boldsymbol{b})$ represents the return obtained by applying portfolio $\boldsymbol{b}$, and a is the desired percentage loss level to be minimized. The first part of the model (4) introduces a downside risk measure that differs from Markowitz's covariance-based risk measure.

These CVaR-based models (Lai et al., 2022; Leung & Wang, 2020; Li, Qin, & Yan, 2022) require the computation of probability inequalities or expectations, which can only be approximated through sampling return data from different time windows. Additionally, these models do not emphasize portfolio sparsity, potentially hindering the generation of sparse portfolio vectors and leading to suboptimal performance for online portfolio trading.

2.5. Asset pre-selection methods

Pre-selecting asset subsets offers precise control over investment scopes while enhancing the diversity of investment strategies (Fulga, Dedu, & Şerban, 2009; Gubu et al., 2021; Hosseinzadeh, Ortobelli Lozza, Hosseinzadeh Lotfi, & Moriggia, 2023; Lorenzo & Arroyo, 2023; Shen & Wang, 2017; Wang, Li, Zhang, & Liu, 2020). For example, Soleymani and Vasighi (2022) employed the k-means++ algorithm to pre-select assets with the highest and lowest risks for portfolio optimization in their study. However, the advantage of this method is primarily applicable in trend situations rather than sideways situations (Han & Ge, 2020). Another approach, proposed by Brito (2023), defines the expected utility, entropy, and variance (EU-EV) model for asset pre-selection.

Combining asset pre-selection methods with portfolio vector optimization leads to a more comprehensive and effective portfolio optimization process. A notable example is the mixed method proposed by Wang et al. (2020), which integrates long short-term memory (LSTM) networks and the mean–variance model for portfolio optimization. In the first stage, LSTM networks are utilized to forecast asset returns and identify assets with higher potential returns. In the subsequent stage, the mean–variance model is employed to optimize the portfolio by considering the selected assets with higher returns.

However, these methods typically have a lower frequency of updating candidate subsets of assets, such as on a monthly or quarterly basis. Moreover, they often lack advanced mechanisms to dynamically switch between different subsets of assets and actively manage downside risk. To address these limitations, this paper introduces a novel minimax optimization model that facilitates the integration of asset pre-selection into the dynamic online portfolio selection.

2.6. Ensemble-based portfolio selection

Ensemble-based portfolio selection, which involves aggregating multiple experts, has shown its effectiveness in enhancing the robustness of portfolio optimization (Dai et al., 2022; He & Yang, 2022; Lin, Zhang, & Yang, 2024; Shavandi & Khedmati, 2022; Tsantekidis et al., 2021; Yang, He, Lin, & Zhang, 2020; Yang, Liu, et al., 2020). For example, He and Yang (2022) proposed an approach that utilizes weak aggregating algorithms to combine expert advice. Another study by Yang, Liu, et al. (2020) employed the Sharpe ratio to automatically select the best-performing agent from an ensemble of proximal policy optimization (PPO), advantage actor–critic (A2C), and deep deterministic policy gradient (DDPG) algorithms. In this method, the three deep reinforcement learning (DRL) experts are retrained every three months. However, a common issue with these ensemble methods is the lack of behavior diversity in expert pool construction, as experts often share similar trading styles due to the use of the same optimization objective or reward function during training. This can lead to a situation where all experts exhibit the same "follow-the-winner" trading behavior (Li & Hoi, 2014).

Several multi-agent deep reinforcement learning (DRL) methods have been proposed to create a rich training environment for an agent making decisions (Chen, Chen, & Huang, 2023; Shavandi & Khedmati, 2022; Tsantekidis et al., 2021). For instance, in the work of Shavandi and Khedmati (2022), multiple deep double-Q learning agents were trained using different time windows, allowing agents indexed by smaller time windows to access actions generated by agents indexed by larger time windows. This approach enables agents to learn long-term trends in advance. Another study by Chen et al. (2023) employed three types of teacher agents to distill knowledge from market environments. The price features generated by these teacher agents were then used as inputs to the student agent's neural network for final trading actions. Additionally, Tsantekidis et al. (2021) trained a student PPO agent from a group of expert PPO agents by augmenting the student's loss with the average of cross-entropy $H[P_i, Q]$, where P_i represents the action distribution of expert i and Q is the action distribution of the student.

These DRL-based methods have certain limitations. Firstly, fixed input dimensions restrict their handling of varying asset numbers during prediction, crucial for online portfolio selection. Secondly, training and fine-tuning DRL-based ensemble algorithms pose challenges, demanding meticulous hyperparameter tuning. Moreover, executing long trading periods, like spanning more than 10 years of daily data, incurs substantial time costs, a challenge for high-frequency portfolio selection.

In addition to using an ensemble of agents to enhance portfolio decision-making robustness, the aggregation of multiple features has been shown to be effective in various studies (Dai et al., 2022; Lai et al., 2022; Shen & Wang, 2017). For example, Dai et al. (2022) introduced the trend peak price tracing (TPPT) method, which utilizes exponential moving averages to aggregate three-state price predictions. Shen and Wang (2017) employed random sampling of asset subsets to improve parameter estimates and integrated the output portfolio vectors from multiple mean–variance optimizers.

We present a concise review of recent advancements in online portfolio selection by summarizing pertinent studies in Table 1. This comparative analysis allows us to emphasize the techniques employed in these studies and draw comparisons to our own research. The table includes important details such as the number of assets (n), cardinality level (c), number of periods (T), epochs for training agents or deep neural networks (e), number of iterations (k), steps within a rollout for DRL inference (τ), and size of ensemble (s). Upon careful examination of the table, it becomes evident that the ASCM framework distinguishes itself as a concise and efficient optimization approach, successfully integrating diversity, sparsity, and risk control with a remarkably low time complexity of $Tn\log n$ for online portfolio selection.

3. Problem setting

In this section, we define the input, output, and goal for trading agents performing online portfolio selection, based on the widely recognized capital growth multiplication model (Cover, 1991; Lai & Yang, 2022; Li et al., 2015).

Table 1
A concise overview of recent research studies in the field of online portfolio selection is presented.

Authors	Diversity			Sparsity		Risk control		Time complexity
	Mean-Variance based optimization	Collection of asset subsets	Ensemble of agents	Asset selection sparsity	Agent selection sparsity	Single-objective	Multi-objective	
Fulga et al. (2009)		✓		✓		✓		$O(Tn)$
Li et al. (2015)				✓		✓		$O(Tn)$
Lai et al. (2017)				✓		✓		$O(Tn^2)$
Shen and Wang (2017)		✓				✓		$O(Tkc^2s)$
Cai and Ye (2019)				✓		✓		$O(Tn)$
Lai et al. (2020)	✓					✓		$O(Tn^2)$
Yang, Liu, et al. (2020)			✓		✓	✓		$O(Tek\tau n^3s)$
Li, Wu, Lu, Wang, and Hu (2020)				✓			✓	$O(Tek2^nn^3)$
Wang et al. (2020)		✓				✓		$O(Tekcn^3)$
Tsantekidis et al. (2021)			✓			✓		$O(Tek\tau n^3s)$
Yin et al. (2021)				✓		✓		$O(Tekn^3)$
Kobayashi et al. (2021)	✓			✓		✓		$O(Tkc\binom{n}{c}n^2)$
Gubu et al. (2021)	✓	✓				✓		$O(Tkc^2)$
Mhammedi and Rakhlin (2022)						✓		$O(T(\log T)^2n^3)$
Li, Uysal, and Mulvey (2022)	✓					✓		$O(Tkn^2)$
Li, Shi, et al. (2022)				✓		✓		$O(Tkcn^2)$
Wei et al. (2022)	✓	✓		✓		✓		$O(Tkcn^4)$
Shi et al. (2022)	✓			✓		✓		$O(Tkcn^2)$
Luo et al. (2022)				✓		✓		$O(Tkn^3\log n)$
He and Yang (2022)			✓					$O(Tn^3)$
Dai et al. (2022)				✓		✓		$O(Tn^2)$
Mazraeh et al. (2022)	✓						✓	$O(Tekn^2)$
Bhatt et al. (2023)			✓					$O(T^{n-1})$
Wu et al. (2024)	✓			✓		✓		$O(Tkn^2)$
Li et al. (2023)	✓			✓		✓		$O(Tkn^2)$
Jalota, Mandal, Thakur, and Mittal (2023)				✓			✓	$O(Tkn^2)$
Hosseinzadeh et al. (2023)		✓				✓		$O(Tkn)$
Chen et al. (2023)			✓			✓		$O(Tek\tau n^3s)$
Lin et al. (2024)			✓			✓		$O(Tekn^2s)$
Fereydooni, Barak, and Sajadi (2024)	✓	✓				✓		$O(Tkn^2)$
Cui, Du, Yang, and Ding (2024)			✓			✓		$O(Tek\tau n^3s)$
Our research		✓		✓			✓	$O(Tn\log n)$

In this model, a trading agent operates within the financial market environment, which encompasses n tradable assets spanning T trading periods. The closing prices of these n assets at the end of each period $t \in [1, T]$ are represented by a price vector $\boldsymbol{p}_t \in \mathbb{R}_+^n$, serving as the standard input for our trading agents.

At the beginning of each period t, a trading agent is assigned the task of making a decision by outputting a portfolio vector $\boldsymbol{b}_t$. This vector represents the distribution of the agent's allocated wealth across the n assets. Mathematically, we define the domain of a portfolio vector $\boldsymbol{b}_t$ as $\boldsymbol{b}_t \in \Delta^{n-1}$, where the symbol Δ^{n-1} denotes the $(n-1)$-dimensional standard unit simplex:

$$\Delta^{n-1} = \left\{ \boldsymbol{b} : \mathbf{1}^\intercal \boldsymbol{b} = 1,\ b_i \geq 0,\ \forall i \in [1, n] \right\}. \tag{5}$$

This domain specification for the portfolio vector $\boldsymbol{b}_t$ imposes two essential constraints: (i) the trading agent must allocate its entire wealth among the n assets at the beginning of each trading period t according to the portfolio vector $\boldsymbol{b}_t$, and (ii) short selling is not allowed for the trading agent. These constraints are commonly referred to as the self-finance property in online portfolio selection (Lai & Yang, 2022; Li & Hoi, 2014).

At the end of each period t, the trading agent is rewarded by the environment based on its action $\boldsymbol{b}_t$ initiated at the beginning of the period t. The reward obtained at the end of period t corresponds to a capital growth rate $\boldsymbol{x}_t^\intercal \boldsymbol{b}_t$, where the relative price vector $\boldsymbol{x}_t$ is defined as follows:

$$\boldsymbol{x}_t := \frac{\boldsymbol{p}_t}{\boldsymbol{p}_{t-1}}. \tag{6}$$

In this equation, $\boldsymbol{x}_t$ is obtained by element-wise division of each pair of elements $p_{t,i}$ and $p_{t-1,i}$ from price vectors $\boldsymbol{p}_t$ and $\boldsymbol{p}_{t-1}$. The relative price vector $\boldsymbol{x}_t$ introduces directionality into the geometric modeling of portfolio decision-making and offers valuable insights into the price movements of individual assets.

The multiplication model of capital growth (Cover, 1991; Lai & Yang, 2022; Li et al., 2015) states that the cumulative reward (cumulative wealth) ω_t received by the trading agent at the end of period t is calculated by multiplying the capital growth rates $\boldsymbol{x}_\tau^\intercal \boldsymbol{b}_\tau$ for each period τ from 1 to t. This can be expressed as follows:

$$\omega_t = \omega_0 \prod_{\tau=1}^{t} \boldsymbol{x}_\tau^\intercal \boldsymbol{b}_\tau, \tag{7}$$

where $\omega_0 > 0$ represents the initial wealth given to the trading agent, typically set to 1.

With this reward mechanism in mind, the goal of the trading agent is to maximize its cumulative reward at the end of each period t by making multi-period decisions on outputting actions $\{\boldsymbol{b}_\tau\}_{\tau=1}^t$. The formal expression of this goal is defined as follows:

$$\max_{\{\boldsymbol{b}_\tau \in \Delta^{n-1}\}_{\tau=1}^t} \prod_{\tau=1}^{t} \boldsymbol{x}_\tau^\intercal \boldsymbol{b}_\tau. \tag{8}$$

Several important points should be considered regarding the goal (8) of the trading agent:

1. The goal (8) cannot be directly employed as an objective function for optimization during each period $\tau \in [1, t]$. This is due to the fact that $\boldsymbol{x}_\tau$ is unknown since the trading agent does not have access to the true price vector $\boldsymbol{p}_\tau$ at the start of period τ.
2. At the start of each period t, the trading agent has access to all previous information, including the price vectors $\{\boldsymbol{p}_\tau\}_{\tau=1}^{t-1}$ and actions $\{\boldsymbol{b}_\tau\}_{\tau=1}^{t-1}$, which can be utilized to predict the new action $\boldsymbol{b}_t$.
3. The goal (8) serves as the widely accepted mathematical model for online portfolio selection and provides a standardized measure for assessing the performance of different approaches (Lai & Yang, 2022; Li & Hoi, 2014).

Based on the analysis presented above, the inputs for online portfolio selection agents consist of a sequence of price vectors $\boldsymbol{p}_t \in \mathbb{R}_+^n$, while the outputs are portfolio vectors $\boldsymbol{b}_t \in \Delta^{n-1}$. The optimization objective of these agents can take various forms, as long as the constraint conditions mentioned in Items 1 and 2 are satisfied. In this paper, we propose the ASCM framework as a solution for constructing online portfolio selection agents. The goal of ASCM agents is to achieve the objective specified in Item 3 by leveraging the diversity, sparsity and risk control provided by the ASCM framework, as depicted in Fig. 1.

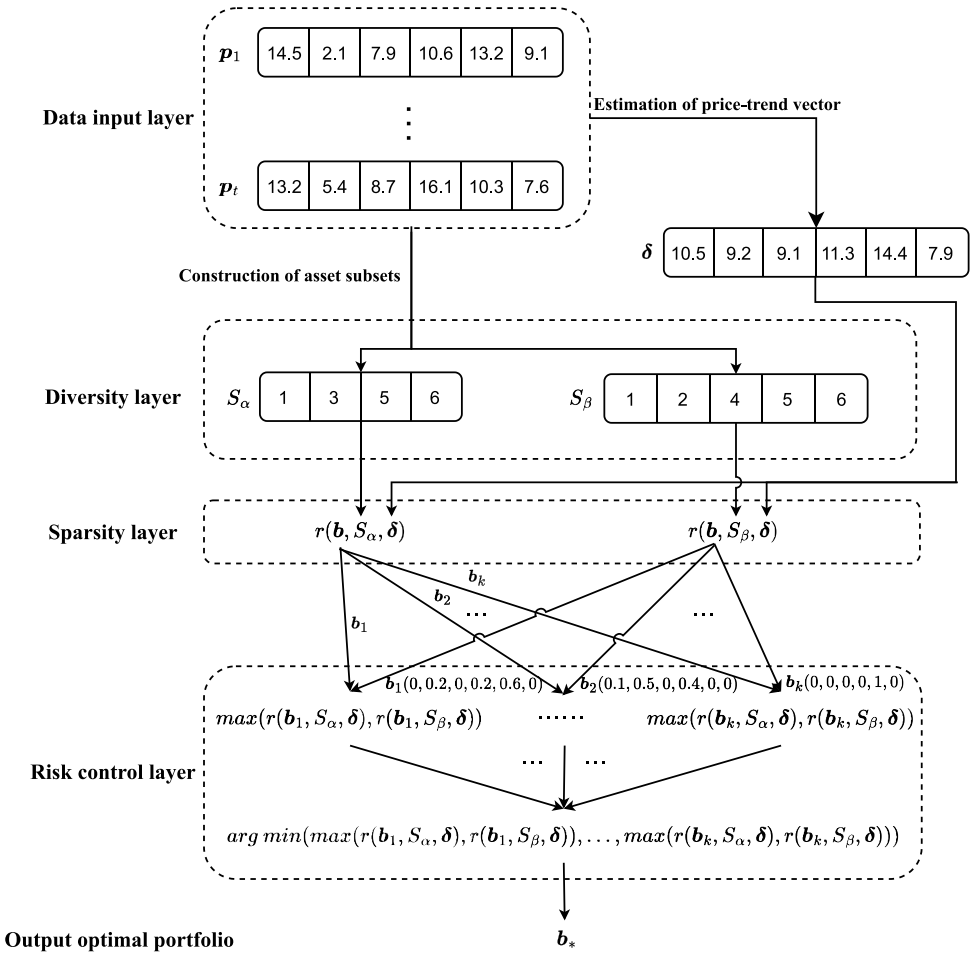


Fig. 1. The architecture of the ASCM framework.

The ASCM framework incorporates successful experiences from various related works, as summarized in Table 1.

Two commonly used benchmark algorithms for evaluating real algorithms in pursuit of the goal (8) are the buy-and-hold (BAH) and the best constant rebalanced portfolio in hindsight (BCRP) (Cover, 1991; Li & Hoi, 2014). The BAH algorithm utilizes the initial wealth ω_0 to invest in a portfolio $\boldsymbol{b}_1$ at the beginning of the first period and does nothing for all the subsequent periods. Consequently, based on Eq. (7), the cumulative wealth ω_n of BAH is given by

$$\omega_n = \omega_0 \boldsymbol{b}_1^\intercal \odot_{t=1}^{T} \boldsymbol{x}_t, \tag{9}$$

where $\odot$ denotes the Hadamard product (element-wise product). A BAH algorithm initialized with the uniform vector $\boldsymbol{b}_1 = \frac{1}{n}\mathbf{1}$ is commonly referred to as the uniform BAH algorithm.

The BCRP algorithm maintains a fixed portfolio $\boldsymbol{b}_t = \boldsymbol{b}_*$ throughout all periods $t \in [1, T]$, where $\boldsymbol{b}_*$ is determined as follows:

$$\boldsymbol{b}_* = \arg\max_{\boldsymbol{b}} \prod_{t=1}^{T} \boldsymbol{x}_t^\intercal \boldsymbol{b}. \tag{10}$$

However, it is worth noting that the BCRP algorithm is impractical since it requires a prior knowledge of all relative price vectors $\{\boldsymbol{x}_t\}_{t=1}^{T}$ in order to determine the optimal portfolio $\boldsymbol{b}_*$.

4. ASCM framework

This section presents the asset subset-constrained minimax (ASCM) optimization framework, which consists of three essential components: the ASCM optimization model, the construction of asset subsets, and the estimation of price-trend vectors. These components are designed to integrate risk control, diversity, and sparsity within a single framework.

4.1. Architecture of ASCM

Fig. 1 illustrates the ASCM framework, which is structured into four conceptual layers: the data input layer, diversity layer, sparsity layer, and risk control layer. Starting from the top, the data input layer receives a sequence of input price vectors $\boldsymbol{p}_t \in \mathbb{R}_+^n$. The data input layer generates two distinct types of data, which are subsequently directed to the diversity layer and sparsity layer, respectively.

The first type of data comprises a collection of asset subsets that are passed to the diversity layer. These subsets are generated by applying specific rules to reduce redundant assets from the entire asset set. In Fig. 1, an example is provided where the collection $\{S_\alpha, S_\beta\}$ represents two distinct investment strategies. It is possible for certain assets to be shared between different investment strategies, such as assets 1, 5, and 6, which are shared between subsets S_α and S_β. Further information regarding the construction of these asset subsets can be found in Section 4.3.

The second type of data comprises the price-trend vectors $\boldsymbol{\delta} \in \mathbb{R}^n$, which are forwarded to the sparsity layer. These vectors capture short-term market trends by approximating price dynamics using first-order and second-order differentials. Further details on the estimation of price trends can be found in Section 4.4.

The diversity layer plays a crucial role in asset pre-selection, as discussed in Section 2.5. Its primary objective is to eliminate redundant assets and form a collection of asset subsets that embody hedge-like investment strategies. The ultimate aim is to offer downside protection within the ASCM framework. In the figure, for the sake of simplicity in implementing ASCM algorithms, it is assumed that the input collection of asset subsets already possesses a desirable hedge structure. As a result, the collection is passed through the sparsity layer without any modifications.

The sparsity layer calculates the potential losses associated with each investment strategy, such as S_α, by evaluating the loss function $r(\boldsymbol{b}_k, S_\alpha, \boldsymbol{\delta})$ for all candidate portfolio vectors $\boldsymbol{b}_k$. These potential losses for the investment strategy S_α are then forwarded to the risk control layer, which plays a crucial role in the final decision-making process. To ensure portfolio sparsity, the candidate portfolio vectors $\boldsymbol{b}_k$ are constrained within a small region determined by the price-trend vector $\boldsymbol{\delta}$. Further information about the initial setting of the candidate portfolio vectors $\boldsymbol{b}_k$ can be found in Sections 4.2 and 6.6.

The risk control layer employs a two-step process to manage risk for using the investment strategies S_α and S_β. In the first step, we calculate and record the maximum losses by comparing each pair of losses, namely $r(\boldsymbol{b}_k, S_\alpha, \boldsymbol{\delta})$ and $r(\boldsymbol{b}_k, S_\beta, \boldsymbol{\delta})$ received from the sparsity layer. These maximum losses serve as reference points for risk assessment. In the second step, we identify the optimal portfolio vector $\boldsymbol{b}_*$ that minimizes the loss among the recorded maximum losses from the first step. This selection process ensures a risk-conscious decision-making approach.

In order to optimize computational efficiency, the sparsity layer and risk control layer have been integrated into the ASCM optimization model, which will be introduced in the upcoming section.

4.2. ASCM optimization model

The ASCM optimization model is designed to generate optimal portfolio vectors by minimizing the maximum risk across multiple investment strategies represented by different asset subsets. This model is formally defined as follows:

Definition 4.1. The asset subset-constrained minimax (ASCM) optimization model

$$\arg\min_{\boldsymbol{b}\in\Delta^{n-1}} \left\{ \max_{i\in[1,m]} r(\boldsymbol{b}, S_i, \boldsymbol{\delta}) \right\} \tag{11}$$

implements the risk control across m investment strategies by minimizing the maximum values of the loss function r within m asset subsets S_i. Here, we assume that $\cap_{i=1}^m S_i \neq \emptyset$.

The loss function $r : \Delta^{n-1} \times 2^n \times \mathbb{R}^n \to \mathbb{R}$ is defined as follows:

$$r(\boldsymbol{b}, S, \boldsymbol{\delta}) = -\boldsymbol{b}^\intercal \boldsymbol{\delta} + \lambda \left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2, \tag{12}$$

where $S \in 2^n$ represents an asset subset, $\boldsymbol{\delta} \in \mathbb{R}^n$ is a user-defined price-trend vector and $\lambda > 0$ is a small preset penalty weight. The first component of the loss function r corresponds to a short-term trend tracking objective, $\boldsymbol{b}^\intercal\boldsymbol{\delta}$, which is shared across multiple strategies S_i. The second component is a regularization term, specifically the Euclidean distance between the portfolio vector $\boldsymbol{b}$ and its projection $\mathcal{P}(S, \boldsymbol{b})$ within the asset subset S.

The projection function $\mathcal{P} : 2^n \times \Delta^{n-1} \to \Delta^{n-1}$ is defined as follows:

$$\mathcal{P}(S, \boldsymbol{b}) = \arg\min_{\boldsymbol{z}\in\mathcal{B}(S)} \left\| \boldsymbol{b} - \boldsymbol{z} \right\|_2, \tag{13}$$

where the set $\mathcal{B}(S)$ is defined by:

$$\mathcal{B}(S) = \{ \boldsymbol{b} : \boldsymbol{b} \in \Delta^{n-1}, b_i = 0 \text{ if } i \notin S \}. \tag{14}$$

The vector set $\mathcal{B}(S)$ represents a subset of the simplex space Δ^{n-1}, where the components b_i with indices not in the set S are constrained to be zero. Therefore, Eq. (14) effectively restricts the candidate investment assets $\boldsymbol{z}$ to belong exclusively to the specified asset subset S.

Since the loss function $r(\boldsymbol{b}, S, \boldsymbol{\delta})$ and the projection function $\mathcal{P}(S, \boldsymbol{b})$ in the optimization problem (11) are non-differentiable with respect to the subset index variable i and the vector $\boldsymbol{b}$, respectively, we utilize the projected subgradient method to solve this problem. In this regard, we begin by introducing the following lemma.

Lemma 1. *There exists a subgradient $\boldsymbol{g} \in \partial r(\boldsymbol{b}, S, \boldsymbol{\delta})$ of the loss function r defined by Eq. (12). This subgradient can be expressed as follows:*

$$\boldsymbol{g} = -\boldsymbol{\delta} + \lambda \frac{\boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b})}{\left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2} \tag{15}$$

Proof. Since all subgradients of the loss function r can be decomposed into two components: $-\boldsymbol{\delta}$ and $\lambda\partial \left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2$, the remaining part of the proof involves deriving a subgradient of $\left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2$, which can be immediately obtained if we can derive a subgradient of $\frac{1}{2} \left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2^2$.

Let $d(\boldsymbol{b}) = \left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2$ and $\phi(\boldsymbol{b}) = \frac{1}{2} d^2(\boldsymbol{b})$, we will show that for any $\boldsymbol{b} \in \mathbb{R}^n$, the gradient $\nabla\phi$ is as follows:

$$\nabla\phi(\boldsymbol{b}) = \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}). \tag{16}$$

To establish Eq. (16), define a function $\varphi_{\boldsymbol{b}}$ by

$$\varphi_{\boldsymbol{b}}(\boldsymbol{c}) = \phi(\boldsymbol{b} + \boldsymbol{c}) - \phi(\boldsymbol{b}) - \boldsymbol{c}^\intercal(\boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b})).$$

Based on the fact that an upper bound of $\varphi_{\boldsymbol{b}}(\boldsymbol{c})$ is

$$\begin{aligned} \varphi_{\boldsymbol{b}}(\boldsymbol{c}) &\leq \frac{1}{2} \left\| \boldsymbol{b} + \boldsymbol{c} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2^2 - \frac{1}{2} \left\| \boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b}) \right\|_2^2 - \boldsymbol{c}^\intercal(\boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b})) \\ &= \frac{1}{2} \left\| \boldsymbol{c} \right\|_2^2, \end{aligned}$$

we can deduce that $\frac{\varphi_{\boldsymbol{b}}(\boldsymbol{c})}{\|\boldsymbol{c}\|_2} \to \boldsymbol{0}$ as $\boldsymbol{c} \to \boldsymbol{0}$. Thus, Eq. (16) is established and the result (15) follows. □

By utilizing Lemma 1 and employing the projected gradient method, we can derive the iterative update rule for the solution variable $\boldsymbol{b}$ in the ASCM optimization model (11). The update rule is presented as follows in Theorem 4.1.

Theorem 4.1. *Let $\{\boldsymbol{b}_k\}_{k\geq 0}$ be the sequence generated by applying the update rule $\boldsymbol{b}_{k+1} = \boldsymbol{b}_k - \alpha_k \boldsymbol{g}_k$, where the step size defined as:*

$$\alpha_k = \begin{cases} \frac{r(\boldsymbol{b}_k, S_{i_k}, \boldsymbol{\delta}) + \|\boldsymbol{\delta}\|_2^2}{\|\boldsymbol{g}_k\|_2^2} & \text{if } \boldsymbol{g}_k \neq \boldsymbol{0} \\ 1 & \text{otherwise} \end{cases} \tag{17}$$

by using the Polyak's step size rule. The subgradient $\boldsymbol{g}_k$ is defined by Lemma 1 and the subset index i_k is defined as $i_k = \arg\max_{i\in[1,m]} r(\boldsymbol{b}_k, S_i, \boldsymbol{\delta})$. Then, we have:

(a) For any $k \geq 0$,

$$\min_{j\in[0,k]} \left\{ \max_{i\in[1,m]} r(\boldsymbol{b}_j, S_i, \boldsymbol{\delta}) \right\} \leq \frac{r(\boldsymbol{b}_0, S, \boldsymbol{\delta}) \left\| \lambda + \boldsymbol{\delta} \right\|_2}{\sqrt{k+1}} - \|\boldsymbol{\delta}\|_2^2, \tag{18}$$

where the subset $S = \cap_{i=1}^m S_i$.

(b) There exist $\boldsymbol{b}_* \in \mathbb{R}^n$ such that $\boldsymbol{b}_k \to \boldsymbol{b}_*$ as $k \to \infty$.

Proof. Let $f(\boldsymbol{b}) = \max_{i\in[1,m]} r(\boldsymbol{b}, S_i, \boldsymbol{\delta})$. According to the convergence of the projected subgradient with Polyak's step size, we have:

$$f(\boldsymbol{b}_k) - f_* \leq \frac{L_f r(\boldsymbol{b}_0, S, \boldsymbol{\delta})}{\sqrt{k+1}} \text{ for any } k \geq 0, \tag{19}$$

where $f_* = \min_{\boldsymbol{b}\in\Delta^{n-1}} f(\boldsymbol{b})$ is a optimal value of the ASCM model (11) and L_f is Lipschitz constant of f. Therefore, we can prove Part (a) by estimating the values of f_* and L_f.

By Eq. (12), it follows that

$$\min_{\boldsymbol{b}} \left\{ \max_i r(\boldsymbol{b}, S_i, \boldsymbol{\delta}) \right\} = \min_{\boldsymbol{b}} \left\{ \max_i \{ -\boldsymbol{b}^\intercal\boldsymbol{\delta} + \lambda \left\| \boldsymbol{b} - \mathcal{P}(S_i, \boldsymbol{b}) \right\|_2 \} \right\}$$

$$\geq \min_{\boldsymbol{b},i} \left\{ -\boldsymbol{b}^\intercal \boldsymbol{\delta} + \lambda \left\| \boldsymbol{b} - \mathcal{P}(S_i, \boldsymbol{b}) \right\|_2 \right\}$$
$$\geq \min_{\boldsymbol{b}} \left\{ -\boldsymbol{b}^\intercal \boldsymbol{\delta} \right\}$$
$$= -\|\boldsymbol{b}\|_2 \|\boldsymbol{\delta}\|_2^2$$
$$\geq -\|\boldsymbol{\delta}\|_2^2 .$$

Hence, we obtain $f_* = -\|\boldsymbol{\delta}\|_2^2$. Let $\boldsymbol{b}, \boldsymbol{d} \in \Delta^{n-1}$. Applying the triangle inequality, we have

$$\begin{aligned} r(\boldsymbol{b}, S, \boldsymbol{\delta}) &= -\boldsymbol{b}^\intercal \boldsymbol{\delta} + \lambda \|\boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b})\|_2 \\ &\leq -\boldsymbol{b}^\intercal \boldsymbol{\delta} + \lambda \|\boldsymbol{b} - \boldsymbol{d}\|_2 + \|\boldsymbol{d} - \mathcal{P}(S, \boldsymbol{d})\|_2 - \boldsymbol{d}^\intercal \boldsymbol{\delta} + \boldsymbol{d}^\intercal \boldsymbol{\delta} \\ &= -\boldsymbol{b}^\intercal \boldsymbol{\delta} + \boldsymbol{d}^\intercal \boldsymbol{\delta} + \lambda \|\boldsymbol{b} - \boldsymbol{d}\|_2 + r(\boldsymbol{d}, S, \boldsymbol{\delta}), \end{aligned}$$

which is the same as

$$\begin{aligned} r(\boldsymbol{b}, S, \boldsymbol{\delta}) - r(\boldsymbol{d}, S, \boldsymbol{\delta}) &\leq -\boldsymbol{b}^\intercal \boldsymbol{\delta} + \boldsymbol{d}^\intercal \boldsymbol{\delta} + \lambda \|\boldsymbol{b} - \boldsymbol{d}\|_2 \\ &= -\boldsymbol{\delta}^\intercal (\boldsymbol{b} - \boldsymbol{d}) + \lambda \|\boldsymbol{b} - \boldsymbol{d}\|_2 . \end{aligned}$$

Thus, we can derive

$$\begin{aligned} |r(\boldsymbol{b}, S, \boldsymbol{\delta}) - r(\boldsymbol{d}, S, \boldsymbol{\delta})| &\leq |-\boldsymbol{\delta}^\intercal (\boldsymbol{b} - \boldsymbol{d}) + \lambda \|\boldsymbol{b} - \boldsymbol{d}\|_2| \\ &\leq \|\boldsymbol{\delta}\|_2 \|\boldsymbol{b} - \boldsymbol{d}\|_2 + \lambda \|\boldsymbol{b} - \boldsymbol{d}\|_2 \\ &= (\lambda + \|\boldsymbol{\delta}\|_2) \|\boldsymbol{b} - \boldsymbol{d}\|_2 . \end{aligned}$$

Given the definition of f, we can deduce that

$$\begin{aligned} |f(\boldsymbol{b}) - f(\boldsymbol{d})| &= \left| \max_{i\in[1,m]} r(\boldsymbol{b}, S_i, \boldsymbol{\delta}) - \max_{i\in[1,m]} r(\boldsymbol{d}, S_i, \boldsymbol{\delta}) \right| \\ &\leq \max_{i\in[1,m]} \left| r(\boldsymbol{b}, S_i, \boldsymbol{\delta}) - r(\boldsymbol{d}, S_i, \boldsymbol{\delta}) \right| \\ &\leq (\lambda + \|\boldsymbol{\delta}\|_2) \|\boldsymbol{b} - \boldsymbol{d}\|_2 . \end{aligned}$$

Consequently, it follows that $L_f = \lambda + \|\boldsymbol{\delta}\|_2$. Part (a) can be derived by substituting the obtained values of f_* and L_f into inequality (19). Since we know $f_* = -\|\boldsymbol{\delta}\|_2^2$, Part (b) is established by taking the limit as k approaches infinity in inequality (18). □

Theorem 4.1 indicates that the iterative process for solving the ASCM model (11) exhibits a sub-linear convergence. This means that given an error bound ϵ for the optimal value f_*, the iterative process can obtain a solution $\boldsymbol{b}_k$ within the time complexity of $O(\frac{1}{\epsilon^2})$, and the error between $f(\boldsymbol{b}_k)$ and f_* is less than or equal to ϵ. In practice, setting an effective initial solution $\boldsymbol{b}_0 = \boldsymbol{e}_j$ from the price-trend vector $\boldsymbol{\delta}$ can greatly accelerate the iterative process. Here, $j = \arg\max_{i\in[1,n]} \delta_i$, and $\boldsymbol{e}_j$ represents the standard basis vector with a value of 1 in the jth component and 0 in all other components. The benefits of this approach are demonstrated in Section 6.6.

Optimal portfolio vectors can be obtained by calling the ASCM model (11) with carefully crafted asset subsets S_i and the price-trend vector $\boldsymbol{\delta}$. The subsequent sections will provide the definitions and elaborate on the details of these two parameters.

4.3. Construction of asset subsets

In this section, we introduce a feature-value clipping method as part of the ASCM framework for constructing asset subsets. This heuristic approach efficiently reduces redundant assets by sampling the two tail portions from the cumulative distribution of price features.

To illustrate the feature-value clipping method, we provide two examples: S_α and S_β. The subset S_α comprises assets exhibiting both high and low price variances, representing contrasting volatility profiles. Conversely, the subset S_β consists of assets with high and low average prices, capturing diverse price levels. These asset subsets serve as representative investment strategy schemes within the ASCM framework, enabling diversification and capturing different market dynamics.

The price feature ξ_i utilized by the asset subset S_α is the sample variance of asset prices observed during recent trading periods. It is defined as follows:

$$\sigma_i^2 = \frac{1}{w_\alpha} \sum_{k=0}^{w_\alpha - 1} (p_{t-k,i} - \mu_i)^2, \tag{20}$$

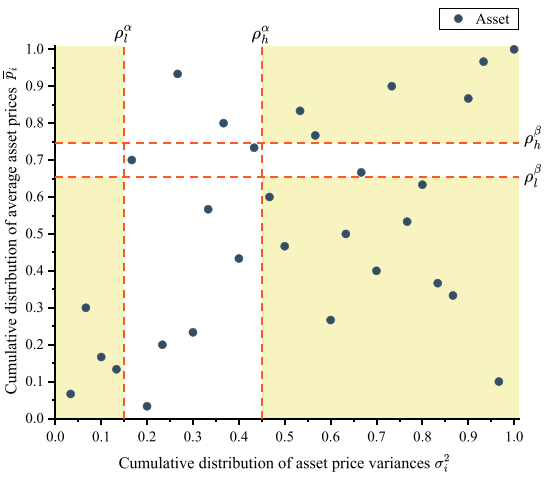


Fig. 2. The construction of asset subsets.

where $p_{t-k,i}$ donates the ith asset price in the $(t-k)$th period and μ_i represents the average price of the ith asset within the window size w_α.

The price feature ξ_i employed by the asset subset S_β is the sample average of asset prices. It is defined as follows:

$$\bar{p}_i = \frac{1}{w_\beta} \sum_{k=0}^{w_\beta - 1} p_{t-k,i}, \tag{21}$$

where w_β is a window size.

The asset subsets S_α and S_β are generated by utilizing the feature-value clipping method, as outlined in Algorithm 1.

Algorithm 1 Asset subset construction algorithm (ASC)

Input:
- The price feature sequence $\{\xi_i \in \mathbb{R}\}_{i=1}^n$ for n assets;
- The proportions $\rho_l, \rho_h \in [0, 1]$ and $\rho_l \leq \rho_h$;

Output:
- The asset index set A

Procedure:
1: Sort $\{\xi_i\}_{i=1}^n$ in ascending order to obtain the sorted feature sequence $\{\xi_{\pi_i}\}_{i=1}^n$;
2: Calculate the low threshold $\xi_l = \xi_{\pi_{\lfloor n*\rho_l \rfloor}}$ and the high threshold $\xi_h = \xi_{\pi_{\lfloor n*\rho_h \rfloor}}$;
3: Output $A = \{i : \xi_i \leq \xi_l \text{ or } \xi_i \geq \xi_h, \forall i \in [1, n]\}$;

Two proportion parameters, ρ_l and ρ_h, are utilized to determine the locations of the two tail portions in the cumulative distribution of the input price feature ξ. The time complexity of Algorithm 1 is $O(n \log n)$, with Step 1 accounting for the majority of the computation time. Here, n represents the number of assets being considered.

Fig. 2 presents an illustrative example of asset subsets S_α and S_β, obtained by executing Algorithm 1 on the 70th period of the Dow Jones Industrial Average (DJIA) dataset (Borodin, El-Yaniv, & Gogan, 2003). The parameter values used for this example are $\rho_l^\alpha = 0.15$, $\rho_h^\alpha = 0.45$, $\rho_l^\beta = 0.654$, and $\rho_h^\beta = 0.734$.

There is a gap of 0.3 between ρ_l^α and ρ_h^α, and a gap of 0.08 between ρ_l^β and ρ_h^β. These gaps indicate the presence of redundant assets that are not essential for computing portfolio vectors. By filtering out 9 assets from the initial set of 30, we obtain a candidate subset $S = S_\alpha \cap S_\beta$ consisting of 21 assets, represented by the yellow region in the figure. Notably, by eliminating the redundant assets, there is a consistent and steady improvement in the performance of ASCM, as observed in Fig. 8(a). A more in-depth discussion on optimizing the proportion settings ρ_l^α, ρ_h^α, ρ_l^β, and ρ_h^β is presented in Section 6.2.

4.4. Estimation of price-trend vectors

The estimation of price-trend vectors $\boldsymbol{\delta}$ in the ASCM model (11) entails two crucial considerations. Firstly, it is essential to incorporate both price state information and price moment information to estimate the price-trend vectors $\boldsymbol{\delta}$. This incorporation allows the ASCM model to effectively utilize these details in optimizing the portfolio return for the next period, as defined by the loss function (12). Secondly, the price-trend vector $\boldsymbol{\delta}$ needs to be projected onto the intersection of multiple subsets S_i, which serve as an input to the ASCM model. This projection is necessary because the first component of the loss function (12) represents a short-term trend tracking objective that is shared among the investment strategies represented by the multiple subsets S_i.

Taking the aforementioned considerations into account, we begin by defining a global price-trend vector $\hat{\boldsymbol{\delta}} \in \mathbb{R}^n$ for all assets. This vector utilizes three estimators to assess short-term price trends, as outlined below:

$$\hat{\delta}_{t,i} = \begin{cases} \dfrac{\max\limits_{0\le k\le w-1} p_{t-k,i}}{p_{t,i}} & \text{if } \Delta p_{t-1,i} \ge 0 \text{ and } \Delta^2 p_{t,i} > 0, \\ \dfrac{\min\limits_{0\le k\le w-1} p_{t-k,i}}{p_{t,i}} & \text{if } \Delta p_{t,i} \ge 0 \text{ and } \Delta^2 p_{t,i} \le 0, \\ \dfrac{\sum_{k=0}^{w-1} p_{t-k,i}}{w p_{t,i}} & \text{otherwise,} \end{cases} \tag{22}$$

where the symbol $\Delta p_{t,i}$ represents an approximation of the first-order finite difference of the prices of the ith asset. It is defined as follows:

$$\Delta p_{t,i} = \frac{1}{5}\sum_{k=0}^{4} p_{t-k,i} - \frac{1}{10}\sum_{k=0}^{9} p_{t-k,i}.$$

Using $\Delta p_{t,i}$, the symbol $\Delta^2 p_{t,i}$ is defined as:

$$\Delta^2 p_{t,i} = \Delta p_{t,i} - \Delta p_{t-1,i}.$$

This represents an approximation of the second-order finite difference of the prices of the ith asset.

For the tth trading period, the global price-trend $\hat{\delta}_{t,i}$ of the ith asset is determined by three price states, as described in Eq. (22). These price states can be visualized as various shapes of a price curve. Specifically, when the first-order difference $\Delta p_{t,i}$ is non-negative and the second-order difference $\Delta^2 p_{t,i}$ is positive, it indicates that the price of the ith asset is following an upward trend with an increasing rate of change, characterized by a convex price curve. This situation typically represents a startup phase with an optimistic market sentiment, and the next-period relative price can be predicted using the first estimator defined in Eq. (22).

Similarly, when the conditions $\Delta p_{t,i} \ge 0$ and $\Delta^2 p_{t,i} \le 0$ are satisfied, it suggests that the price of the ith asset is following an upward trend but with a decreasing rate of change, characterized by a concave price curve. This situation typically represents a maturing or consolidation phase in an overbought market sentiment, and the next-period relative price can be predicted using the second estimator defined Eq. (22).

In addition to the two typical market trends mentioned above, we incorporate the mean relative price $\frac{\sum_{k=0}^{w-1} p_{t-k,i}}{w p_{t,i}}$ as the estimator for the third case in Eq. (22). This estimator accounts for scenarios where the price trend is not clearly increasing or decreasing, representing a neutral or sideways market sentiment. The utilization of mean relative price is a common practice in trend-reversion type algorithms like OLMAR (Li et al., 2015) and robust median reversion (RMR) (Huang, Zhou, Li, Hoi, & Zhou, 2016).

To demonstrate the effectiveness of the three global price-trend estimators defined in Eq. (22), we select a subset of price data from the DJIA dataset spanning from period 13 to period 49. Within this subset, we specifically focus on six periods: 25, 28, 33, 37, 40, and 47, which are identified for detailed analysis and visualization in Fig. 3. These six periods represent a typical micro-market cycle of price trends, allowing us to evaluate the performance of the proposed estimators under various market scenarios.

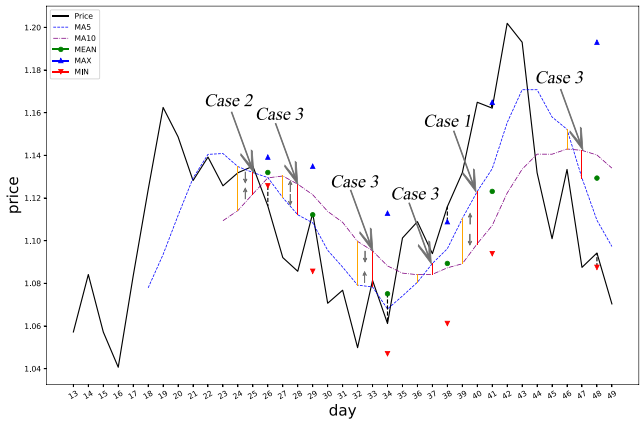


Fig. 3. Visual representation of three global price-trend estimators as defined in Eq. (22).

We compute the three conditions and their corresponding estimators using Eq. (22) for the six periods and represent them visually on the figure. The figure demonstrates that these estimators provide accurate predictions for the upcoming periods, specifically periods 26, 29, 34, and 41. For instance, during period 28, the condition corresponds to the third case indicated as "Case 3" in the figure. In the subsequent period, the third estimator, represented by the green point, precisely aligns with the actual price curve denoted by the black color.

Once the global price-trend vectors $\hat{\boldsymbol{\delta}}$ have been estimated using Eq. (22), the next step is to project them onto the intersection of multiple asset subsets S_i in order to obtain the price-trend vectors $\boldsymbol{\delta}$. This projection is performed according to the following rule:

$$\delta_{t,i} = \begin{cases} \hat{\delta}_{t,i} & \text{if } i \in S \\ 0 & \text{otherwise,} \end{cases} \tag{23}$$

where the intersection $S = \cap_{i=1}^{m} S_i$ represents the shared elements among the asset subsets S_i. By filtering out the price trends of assets that do not belong to this intersection, we successfully eliminate their influence on the first term $-\boldsymbol{b}^\intercal\boldsymbol{\delta}$ in the loss function (12). Note that this term serves as a short-term trend tracking objective shared specifically among the asset subsets S_i, rather than the entire set of assets.

5. ASCM algorithm

In this section, we present the ASCM algorithm as a manifestation of the ASCM framework. The ASCM algorithm utilizes price-trend vectors $\boldsymbol{\delta}$ in conjunction with asset subsets S_α and S_β to invoke the ASCM model (11) and derive optimal portfolio vectors $\boldsymbol{b}$ for investment purposes. The detailed description of the ASCM algorithm can be found in Algorithm 2.

In Step 5, the two function calls using the ASC function indicate that we obtain the two asset subsets S_α and S_β by invoking the ASC Algorithm 1 with their respective parameters.

In Step 6, we verify whether there is a non-empty intersection between S_α and S_β, as required by the ASCM model (11). If the intersection is found to be empty, we proceed to include the asset with the minimum price variance into both S_α and S_β, as demonstrated in Steps 7 and 8.

Therefore, Steps 6–8 guarantee the presence of at least one feasible solution when solving the ASCM model with S_α and S_β, while minimizing the impact on the original investment strategies represented by S_α and S_β. The asset with the minimum price variance exhibits a high level of price stability and can be considered a conservative candidate for investment purposes in cases where there is no consensus between S_α and S_β.

Algorithm 2 ASCM algorithm.

Input:

- The price-vector sequence $\{\boldsymbol{p}_i \in \mathbb{R}_+^n\}_{i=1}^T$ of n assets;
- The proportions ρ_l^α, ρ_h^α, ρ_l^β, $\rho_h^\beta \in [0,1]$;
- The window sizes w_α, w_β and w;
- The penalty weight $\lambda > 0$ and the error bound $\epsilon > 0$;

Output:

- The portfolio vectors $\boldsymbol{b}_t \in \Delta^{n-1}$ for periods $t = 1, ..., T$.

Procedure:

1: Initialize and output $\boldsymbol{b}_t = \frac{1}{n}\mathbf{1}$ for $t = 1, ..., w$;
2: **for** $t = w, ..., T-1$ **do**
3: Compute the price variances $\{\sigma_i^2\}_{i=1}^n$ using Eq. (20);
4: Compute the average prices $\{\overline{p}_i\}_{i=1}^n$ using Eq. (21);
5: Let $S_\alpha = \mathrm{ASC}(\{\sigma_i^2\}_{i=1}^n, \rho_l^\alpha, \rho_h^\alpha)$, and $S_\beta = \mathrm{ASC}(\{\overline{p}_i\}_{i=1}^n, \rho_l^\beta, \rho_h^\beta)$;
6: **if** $S_\alpha \cap S_\beta = \varnothing$ **then**
7: Let $i = \mathrm{argmin}_{i\in[1,n]}\, \sigma_i^2$;
8: Let $S_\alpha = S_\alpha \cup \{i\}$, and $S_\beta = S_\beta \cup \{i\}$;
9: $S = S_\alpha \cap S_\beta$;
10: Compute the price-trend vector $\boldsymbol{\delta}$ using Eq. (22) and Eq. (23);
11: Let $k = 0$, and $\hat{\boldsymbol{b}}_0 = \boldsymbol{e}_j$, where $j = \mathrm{argmax}_{i\in[1,n]}\, \delta_i$;
12: **while** $k = 0$ or $\left\|\hat{\boldsymbol{b}}_k - \hat{\boldsymbol{b}}_{k-1}\right\|_2 > \epsilon$ **do**
13: Let $i_k = \mathrm{argmax}_{i\in[\alpha,\beta]} r(\hat{\boldsymbol{b}}_k, S_i, \boldsymbol{\delta})$;
14: Compute the subgradient $\boldsymbol{g}_k$ using S_{i_k} through Eq. (15);
15: Update $\hat{\boldsymbol{b}}_{k+1} = \hat{\boldsymbol{b}}_k - \alpha_k \boldsymbol{g}_k$ using Eq. (17);
16: $k = k + 1$;
17: **end while**
18: Output $\boldsymbol{b}_{t+1} = \hat{\boldsymbol{b}}_k$;
19: **end for**

Algorithm 2 uses a greedy initialization method in Step 11 to efficiently solve the ASCM model (11). This sets the initial feasible solution $\hat{\boldsymbol{b}}_0$ to $\boldsymbol{e}_j$, where asset j has the maximum price trend. This approach effectively accelerates the convergence of the iteration in Steps 13–16, resulting in an approximate $O(1)$ time complexity. Our experiments in Section 6.6 validate this fast convergence enabled by the greedy initialization.

To analyze the time complexity of Algorithm 2 for each trading period t, we can break it down as follows:

- Computing the price features $\{\sigma_i^2\}_{i=1}^n$ and $\{\overline{p}_i\}_{i=1}^n$ for all assets in Steps 3 and 4 takes $O(n(w_\alpha + w_\beta))$ time.
- Constructing the asset subsets S_α and S_β using the ASC Algorithm 1 in Step 5 requires $O(n \log n)$ time.
- Determining the intersection of subsets S_α and S_β, computing the price-trend vector $\boldsymbol{\delta}$, and setting the initial feasible solution $\hat{\boldsymbol{b}}_0$ in Steps 9–11 have a time complexity of $O(nw)$.
- The iterative process in Steps 13–16 has an approximate time complexity of $O(1)$ when leveraging our greedy initialization method.

Therefore, the overall time complexity of Algorithm 2 is $O(Tn(w_\alpha + w_\beta + w + \log n))$, where T denotes the number of trading periods, n represents the number of assets, and w_α, w_β and w represent different window sizes of periods. The most time-consuming step is the construction of subsets using the ASC Algorithm 1.

6. Experiments

This section presents two types of experiments that evaluate the ASCM algorithm. The first set of experiments explores key characteristics of ASCM, including diversity, risk control, sparsity, and optimization efficiency in solving the ASCM model. The second set of experiments compares ASCM's performance with representative algorithms using six real-world financial market datasets.

Table 2
Summary of testing datasets.

Dataset	Region	Time	Days	Assets
DJIA	US	01/Jan/2001–14/Jan/2003	507	30
SP500	US	02/Jan/1998–31/Jan/2003	1276	25
TSE	CA	04/Jan/1994–31/Dec/1998	1259	88
HS300	CN	21/Jan/2016–16/Oct/2017	421	44
STOXX50	EU	22/May/2001–11/Apr/2016	3885	49
CRYPTO	Global	01/Jan/2020–31/Dec/2022	1096	23

6.1. Datasets

The experimental data encompass six datasets, five of which have gained significant prominence within the online portfolio selection community. The first three datasets, namely DJIA (Borodin et al., 2003), SP500 (Borodin et al., 2003), and TSE (Borodin et al., 2003), have been derived from reputable stock market indices, including the Dow Jones Industrial Average, Standard & Poor's 500, and Toronto Stock Exchange, respectively. These datasets offer a rich representation of the dynamics and trends observed in major financial markets.

In addition to these established datasets, our research incorporates two distinct datasets: HS300 (Lai, Dai, Ren, & Huang, 2018) and STOXX50 (Carleo, Cesarone, Gheno, & Ricci, 2017). These datasets have been meticulously collected from the China Stock Index 300 and the Euro Stoxx 50 Market Index, respectively. By including these datasets, we aim to capture the unique characteristics and dynamics of these specific regional markets.

Moreover, we introduce a novel dataset named "CRYPTO", which provides a carefully curated collection of price sequences for 23 cryptocurrencies. This dataset spans from January 1, 2020, to December 31, 2022.[2] By incorporating the rapidly evolving cryptocurrency market, we aim to explore the challenges and opportunities presented by this emerging asset class.

To facilitate a deeper understanding of these datasets, we provide a detailed overview of their basic information in Table 2. This includes key statistics, such as the number of assets, the time period covered, and any additional features or characteristics specific to each dataset.

6.2. Parameter setting of ASCM

In this section, we present an effective heuristic method for establishing the default configuration for the ASCM algorithm, which encompasses the parameters defined in Algorithm 2. Moreover, to explore and potentially identify optimal parameter configurations for the task, advanced search techniques like Bayesian optimization can be utilized.

We initially establish three parameters with their default values based on prior research. Specifically, the error bound ϵ in Algorithm 2 is set to 5×10^{-4}, which effectively limits the iterative error within Step 12 of Algorithm 2. The window sizes w_β and w, employed for computing average asset prices (Eq. (21)) and price-trend vectors (Eq. (22)) respectively, are both assigned the value of 5. These default values are recommended by previous studies such as Lai et al. (2017), Li et al. (2015).

Subsequently, we ascertain the parameter settings for the remaining six parameters by assessing the performance of ASCM concerning each of these parameters on a random 10% sample from the testing datasets. This process is carried out through the following steps:

1. Determine the window size w_α while keeping the proportions ρ_l^α and ρ_l^β fixed at 0.3, the proportions ρ_h^α and ρ_h^β fixed at 0.4, and the penalty weight λ fixed at 2×10^{-4}. The performance of ASCM in this specific configuration is illustrated in Fig. 4.

[2] The CRYPTO dataset is sourced from https://finance.yahoo.com/.

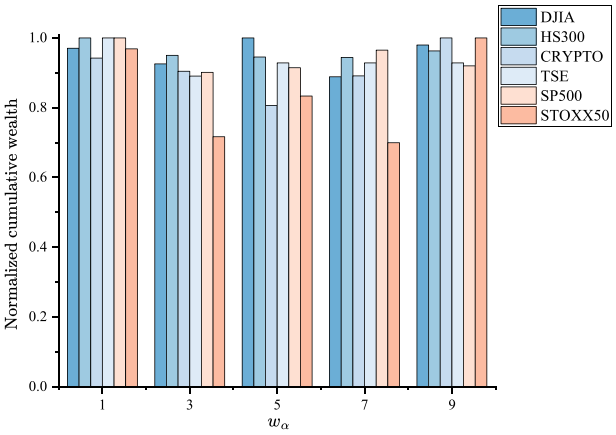


Fig. 4. Cumulative wealth of ASCM with varying w_α.

2. Determine the proportions ρ_l^α and ρ_h^α using w_α obtained in Step 1 while keeping the other parameters at the same setting as Step 1. The performance of ASCM in this specific configuration is shown in Fig. 5.
3. Determine the proportions ρ_l^β and ρ_h^β using w_α obtained in Step 1 and ρ_l^α and ρ_h^α obtained in Step 2 while keeping the penalty weight λ fixed at 2×10^{-4}. The performance of ASCM in this specific configuration is depicted in Fig. 6.
4. Determine the penalty weight λ using w_α, ρ_l^α, ρ_h^α, ρ_l^β and ρ_h^β obtained in Step 1, 2 and 3. The performance of ASCM in this specific configuration is presented in Table 3.

Note that the proportions ρ_l^α, ρ_h^α, ρ_l^β, and ρ_h^β are defined within the range of $[0, 1]$, as specified in Algorithm 1. Therefore, in Step 1, we aim to find the optimal window size w_α by considering only the assets whose price variances and average prices fall below 30% and exceed 40% in their respective sample cumulative distributions.

In order to determine the optimal window size w_α and the proportions ρ_l^α, ρ_h^α, ρ_l^β, and ρ_h^β, we initially assign a small value of 2×10^{-4} to the penalty weight λ. This choice introduces a minor adjustment to the main objective $-\boldsymbol{b}^\intercal \boldsymbol{\delta}$ in the loss function (12), which contributes to maintaining the stability when evaluating the performance of ASCM, particularly in comparison to using a larger value of λ.

In Step 1, Fig. 4 demonstrates that choosing $w_\alpha = 9$ is a superior option compared to $w_\alpha = 1$ even though the latter displays a more balanced performance across all datasets. The preference for a larger window size arises from the fact that w_α is utilized in calculating asset price variances, as described in Eq. (20). By selecting a larger window size, we attain more reliable and stable estimations of asset variances.

In Step 2, Fig. 5 displays the distributions of the cumulative wealth (CW) of ASCM across different settings of ρ_l^α and ρ_h^α. The observed sparse modes indicate that points with high CWs tend to converge towards high-density regions, where ρ_h^α predominantly falls within the range of [0.4, 0.5] across most of the testing datasets. This suggests the existence of a relatively stable percentage of assets that can be excluded to achieve optimal performance in ASCM. To ensure comprehensive coverage, we choose the setting $\rho_l^\alpha = 0.15$ and $\rho_h^\alpha = 0.45$ as a relatively conservative configuration for ASCM.

In Step 3, Fig. 6 illustrates ASCM's CW distributions across various ρ_l^β and ρ_h^β settings. These distributions exhibit similar sparse modes as observed in Fig. 5. Particularly, a dense region is noticeable within the interval of $[0.6, 0.8]$. Within this high-density region, the discrepancies between ρ_h^β and ρ_l^β are relatively smaller compared to those of ρ_h^α and ρ_l^α. For instance, on the TSE dataset, the gap measures approximately 0.1 when $\rho_h^\beta = 0.7$. To determine suitable parameter values, we perform a simple grid search in the vicinity of this region and find that the setting $\rho_l^\beta = 0.645$ and $\rho_h^\beta = 0.735$ yield favorable outcomes for the majority of cases.

Table 3
Cumulative wealth of ASCM with varying λ.

λ	DJIA	HS300	CRYPTO	TSE	SP500	STOXX50
0	1.16	0.92	185.89	2.28	**1.60**	**0.97**
2×10^{-4}	1.16	0.92	185.90	2.28	**1.60**	**0.97**
2×10^{-3}	1.16	0.92	**186.61**	2.28	**1.60**	**0.97**
2×10^{-2}	**1.17**	0.90	136.02	**2.31**	1.48	0.95
2×10^{-1}	1.02	1.01	60.22	1.59	1.37	0.96
1	0.88	**1.05**	2.75	1.00	1.32	0.66

In Step 4, we execute the ASCM algorithm while varying the parameter λ after obtaining the parameter settings of $w_\alpha = 9$, $\rho_l^\alpha = 0.15$, $\rho_h^\alpha = 0.45$, $\rho_l^\beta = 0.645$, and $\rho_h^\beta = 0.735$. The resulting CWs are recorded in Table 3. Notably, the CWs of ASCM demonstrate minimal variation within the range of $[2 \times 10^{-4}, 2 \times 10^{-3}]$, indicating a stable performance. However, when λ equals or exceeds 2×10^{-2}, the performance of ASCM starts to decline compared to when λ is less than 2×10^{-2}. This observation suggests that assigning a weight higher than 2×10^{-2} to the term $\|\boldsymbol{b} - \mathcal{P}(S, \boldsymbol{b})\|_2$ in Eq. (12) can impact the performance of ASCM. Therefore, we choose 2×10^{-3} as the default setting for the parameter λ.

In summary, the default configuration of the ASCM algorithm utilizes the following parameter settings: $w_\alpha = 9$, $\rho_l^\alpha = 0.15$, $\rho_h^\alpha = 0.45$, $\rho_l^\beta = 0.645$, $\rho_h^\beta = 0.735$, and $\lambda = 2 \times 10^{-3}$.

6.3. Diversified investment strategies based on asset subsets

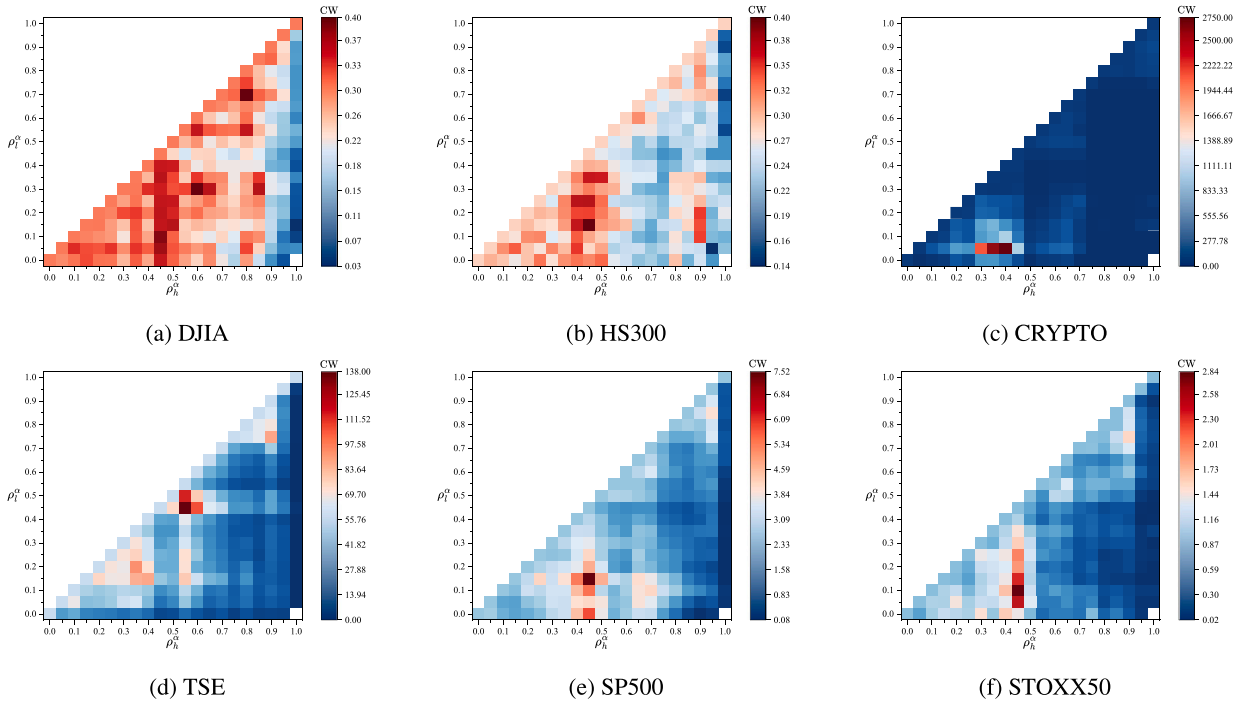
In this section, we delve into the analysis of the ASCM framework's capabilities in facilitating diverse investment strategies through the combination of various asset subsets.

Within the ASCM framework, each investment strategy is characterized by a unique collection of asset subsets. For instance, $\{S_\alpha, S_\beta\}$ represents one investment strategy, while $\{S, S_\alpha\}$ represents another. The cumulative wealth generated by applying an investment strategy, as defined by Eq. (7), can be obtained by making a small modification to the ASCM algorithm (Algorithm 2). This modification involves replacing Step 5 with the corresponding collection of asset subsets, while keeping the parameter settings the same as defined in Section 6.2. For instance, by using the collection $\{S, S_\alpha\}$ instead of $\{S_\alpha, S_\beta\}$, a new cumulative wealth is obtained from the investment strategy $\{S, S_\alpha\}$. The cumulative wealth achieved by nine combinations of asset subsets is presented in Table 4.

These combinations of investment strategies encompass the entire asset set S, the asset subset S_α, the asset subset S_β, as well as their respective complements $\overline{S}_\alpha = S \backslash S_\alpha$ and $\overline{S}_\beta = S \backslash S_\beta$. Notably, the table emphasizes the highest cumulative wealth achieved for each dataset, prominently displayed in bold black font.

The ASCM framework effectively supports diversified investment strategies represented by collections of asset assets. For instance, on the DJIA dataset, the cumulative wealth of investment strategies $\{S_\alpha, S_\beta\}$ and $\{S_\alpha, \overline{S}_\beta\}$ differs by a factor of 5.11. Similarly, on the CRYPTO dataset, the disparity reaches a factor of 2524.58, even though the two strategies differ by only one subset. This highlights the significance of diversified strategies, as the ASCM model (11) excels at differentiating among various collections of asset subsets.

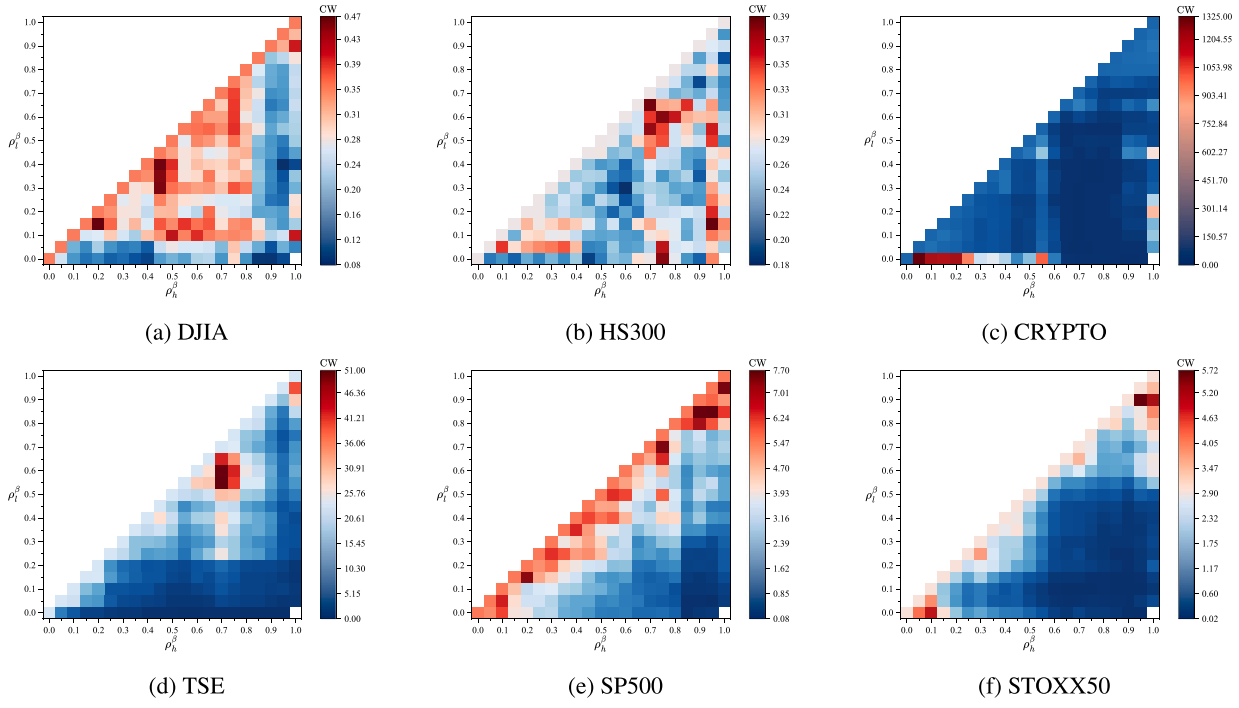
Furthermore, we calculate the mean and variance of the cumulative wealth generated by applying these investment strategies to each dataset. The results demonstrate notable volatility in the cumulative wealth variances across different datasets, as indicated in the bottom line of Table 4. These findings underscore the substantial differences among these markets, highlighting the influential role of asset subsets as controlling factors in representing diverse investment strategies.



(a) DJIA (b) HS300 (c) CRYPTO

(d) TSE (e) SP500 (f) STOXX50

Fig. 5. Cumulative wealth of ASCM with varying ρ_l^α and ρ_h^α.



(a) DJIA (b) HS300 (c) CRYPTO

(d) TSE (e) SP500 (f) STOXX50

Fig. 6. Cumulative wealth of ASCM with varying ρ_l^β and ρ_h^β.

The investment strategy $\{S_\alpha, S_\beta\}$ consistently emerges as the top-performing strategy across all datasets, except for the TSE dataset. Based on this exceptional performance, the ASCM algorithm adopts the strategy $\{S_\alpha, S_\beta\}$ as the default choice, as illustrated in Algorithm 2.

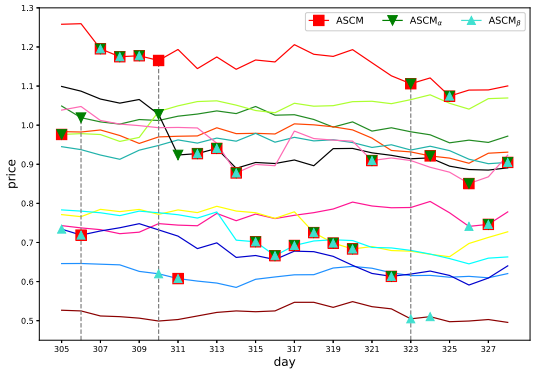
6.4. Analysis of risk control behaviors

In this section, we analyze the risk control behaviors exhibited by the ASCM algorithm. We achieve this by carefully observing and analyzing its asset selection process. We employ a reference-algorithm analysis method on a representative sample dataset to facilitate our analysis.

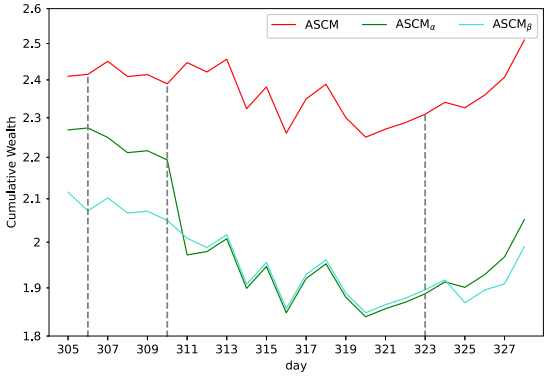
Given that the ASCM algorithm adopts the asset subsets $\{S_\alpha, S_\beta\}$ as its investment strategy, with risk control in the ASCM model (11) focusing on minimizing the maximum risk across the subsets S_α and S_β, we introduce two reference algorithms, ASCM$_\alpha$ and ASCM$_\beta$, to analyze the risk control behavior of the ASCM algorithm. The algorithms ASCM$_\alpha$

Table 4
Cumulative wealth of different investment strategies.

Strategy	DJIA	HS300	CRYPTO	TSE	SP500	STOXX50
$\{S, S_\alpha\}$	3.69	1.54	1788.51	278.40	68.35	17.31
$\{S, S_\beta\}$	3.31	1.66	759.3	**702.93**	28.90	9.34
$\{S, \overline{S}_\alpha\}$	0.95	0.88	2.14	2.44	1.00	0.71
$\{S, \overline{S}_\beta\}$	0.71	1.28	1.25	0.36	1.17	0.28
$\{S_\alpha, S_\beta\}$	**4.14**	**2.05**	**3155.73**	446.22	**75.08**	**24.22**
$\{S_\alpha, \overline{S}_\beta\}$	0.81	0.93	1.25	0.55	1.01	0.28
$\{\overline{S}_\alpha, S_\beta\}$	1.24	0.81	2.07	2.72	0.84	0.73
$\{\overline{S}_\alpha, \overline{S}_\beta\}$	0.72	1.39	1.20	1.69	1.73	1.16
$\{S, S\ \}$	3.96	1.30	283.81	483.55	31.27	8.17
Average	2.17	1.32	666.00	213.21	23.26	6.86
Variance	2.13	0.15	1091201.71	66 161.44	806.57	68.75



(a) Asset selection on the DJIA dataset.



(b) Cumulative wealth on the DJIA dataset.

Fig. 7. Asset selection process on the DJIA dataset: analysis and results.

and ASCM$_\beta$ utilize the asset subsets $\{S, S_\alpha\}$ and $\{S, S_\beta\}$ respectively as their investment strategies.

Fig. 7 visually illustrates the asset selection process employed by the ASCM, ASCM$_\alpha$, and ASCM$_\beta$ algorithms on the DJIA dataset. In particular, Fig. 7(a) displays the assets chosen by these algorithms during each trading period. To facilitate the analysis of asset selection behavior, Fig. 7(b) depicts the corresponding dynamics of cumulative wealth, aligned with the timeline presented in Fig. 7(a).

During specific trading periods, indicated by dashed lines in the figures, the ASCM algorithm diverges from ASCM$_\alpha$ and ASCM$_\beta$ in asset selection, even though these three algorithms generally exhibit similar selections. For example, at period 310, the asset chosen by ASCM exhibits an increase, while the assets selected by ASCM$_\alpha$ and ASCM$_\beta$ both show a decrease in the following period. Similar asset selection behaviors of ASCM can also be observed at periods 306 and 323. These findings underscore the superior risk control capability of ASCM compared to ASCM$_\alpha$ and ASCM$_\beta$ in avoiding downside risk through the selection of well-performing assets during crucial trading periods. Consequently, the ASCM algorithm achieves significantly higher cumulative wealth.

The ASCM algorithm's superior risk control compared to ASCM$_\alpha$ and ASCM$_\beta$ can be attributed to its utilization of both the asset subsets S_α and S_β. Specifically, ASCM constructs price-trend vectors $\boldsymbol{\delta}$ by taking the intersection of S_α and S_β, as shown in Eq. (23), whereas ASCM$_\alpha$ and ASCM$_\beta$ construct $\boldsymbol{\delta}$ using only S_α or S_β, respectively. This enables the ASCM algorithm to effectively filter out assets with weaker growth trends, enhancing its risk control capability.

6.5. Controllable sparse portfolios

This section showcases the flexibility of the ASCM algorithm in generating portfolio vectors with varying levels of sparsity through the adjustment of the penalty weight λ in Eq. (12). It is important to note, as discussed in Section 1, that the degree of sparsity in the portfolio vectors plays a crucial role in determining the performance of a portfolio algorithm.

We performed several runs of the ASCM algorithm, varying the penalty weight λ, to assess sparsity and performance. The results are summarized in Table 5, where the optimal values for each dataset are highlighted in bold black fonts.

The sparsity metric $\|\boldsymbol{b}\|_0$ represents the l_0-norm of the portfolio vectors $\boldsymbol{b}$ generated by ASCM, indicating the number of non-zero components in $\boldsymbol{b}$. For example, on the DJIA dataset with the setting $\lambda = 2 \times 10^{-3}$, the ASCM algorithm selected only one asset in 445 out of a total of 507 periods, achieving a cumulative wealth (CW) of 4.14 and a maximum drawdown (MDD) of 29.79%. As the ASCM algorithm initially sets the portfolio vectors as $\frac{1}{m}\mathbf{1}$ for the first five days, there are only five non-sparse portfolio vectors with the setting $\lambda = 0$.

The number of sparse portfolio vectors produced by ASCM decreases as the penalty weight λ increases. When λ is less than $2{\times}10^{-3}$, ASCM can generate sufficiently sparse portfolio vectors while maintaining high returns. However, when λ exceeds 2×10^{-2}, ASCM cannot produce sparse enough portfolio vectors, greatly impacting its performance.

Maintaining an appropriate level of sparsity in the portfolio vectors generated by the ASCM algorithm is crucial, rather than solely pursuing absolute sparsity. This is evident when comparing the cumulative wealth obtained under the absolute sparsity setting of $\lambda = 0$ with that achieved at a specific level of sparsity in the setting of $\lambda = 2 \times 10^{-3}$ on the SP500 and STOXX50 datasets, respectively.

6.6. Greedy initialization method for solving ASCM model

In this section, we evaluate the effectiveness of the greedy initialization method utilized by the ASCM algorithm (Algorithm 2) for setting the initial feasible solution $\hat{\boldsymbol{b}}_0$ in solving the ASCM model (11).

Table 5
Exploring the influence of penalty weights λ on sparsity and performance.

λ	Metric	DJIA	HS300	CRYPTO	TSE	SP500	STOXX50
$\lambda = 0$	$\|\boldsymbol{b}\|_0 = 1$	**502**	**416**	**1091**	**1254**	**1271**	**3880**
	$\|\boldsymbol{b}\|_0 > 1$	**5**	**5**	**5**	**5**	**5**	**5**
	CW	**4.14**	**2.05**	3148.69	**446.46**	74.72	23.57
	MDD (%)	29.81	24.40	87.24	54.65	39.93	57.26
$\lambda = 2 \times 10^{-3}$	$\|\boldsymbol{b}\|_0 = 1$	445	322	918	1123	1097	3190
	$\|\boldsymbol{b}\|_0 > 1$	62	99	178	136	179	690
	CW	**4.14**	**2.05**	3155.81	446.22	**75.08**	**24.22**
	MDD (%)	**29.79**	24.40	87.20	54.65	39.92	56.97
$\lambda = 2 \times 10^{-2}$	$\|\boldsymbol{b}\|_0 = 1$	177	57	474	609	420	874
	$\|\boldsymbol{b}\|_0 > 1$	330	359	622	650	856	3011
	CW	3.94	1.83	**3233.09**	335.11	63.60	18.91
	MDD (%)	29.80	22.56	83.41	57.41	42.66	56.60
$\lambda = 2 \times 10^{-1}$	$\|\boldsymbol{b}\|_0 = 1$	10	4	37	40	11	19
	$\|\boldsymbol{b}\|_0 > 1$	497	417	1059	1219	1265	3865
	CW	1.83	1.53	1998.93	95.61	9.83	5.42
	MDD (%)	31.05	14.69	**64.00**	51.41	28.36	**48.05**
$\lambda = 1$	$\|\boldsymbol{b}\|_0 = 1$	0	0	0	0	0	0
	$\|\boldsymbol{b}\|_0 > 1$	507	421	1096	1259	1276	3885
	CW	0.70	1.49	27.63	5.57	2.60	2.09
	MDD (%)	45.72	**10.72**	67.11	**26.39**	**28.19**	53.48

To ensure a comprehensive and comparative analysis, we compare the performance of the greedy initialization method with alternative initialization approaches on multiple datasets.

Specifically, we execute the ASCM algorithm using three distinct initialization methods for Step 11 of Algorithm 2, respectively. We then measure the number of iterations performed from Step 13 to Step 16 in Algorithm 2 for each initialization method. The three initialization methods employed are as follows:

- Greedy initialization method (GIM): In this method, we set $\hat{\boldsymbol{b}}_0 = \boldsymbol{e}_j$, where $j = \arg\max_{i \in [1,n]} \delta_i$;
- Random initialization method (RIM): In this method, we set $\hat{\boldsymbol{b}}_0 = \boldsymbol{e}_i$, where i is randomly chosen from the set $\{1, \ldots, n\}$;
- Average initialization method (AIM): In this method, we set $\hat{\boldsymbol{b}}_0 = \frac{1}{n}\mathbf{1}$.

Since the number of iterations executed from Step 13 to Step 16 in Algorithm 2 varies as a random variable in each trading period, we provide an approximation estimation by presenting the top three most frequently occurring numbers of iterations in Table 6. For example, when using GIM, the number of iterations is 1 for 451 periods, which corresponds to 89.94% of the total periods in the DJIA dataset.

The advantage of using GIM over the RIM and AIM methods is evident, as the approximation averages of its number of iterations is consistently low across six datasets, with values of 1.17, 1.09, 1.08, 1.15, 1.18, and 1.21 respectively. In contrast, the numbers of iterations using RIM and AIM span a wide range without specific modes. This result strongly indicates that the time complexity of executing instructions from Step 13 to Step 16 in Algorithm 2, with the GIM setting at Step 11, is approximately $O(1)$. This complexity is considerably lower than the theoretical upper bound of $O(\frac{1}{\epsilon^2})$. These findings highlight the efficiency and superior performance of using GIM compared to alternative methods, namely RIM and AIM.

The effectiveness of the ASCM algorithm using GIM can be attributed to two key factors: (i) By utilizing the price-trend vector $\boldsymbol{\delta}$, the assets considered for the initial vector $\hat{\boldsymbol{b}}_0$ are restricted to the intersection of asset subsets S_α and S_β. As a result, the need to search for assets across the entire asset set is significantly reduced. (ii) The ASCM algorithm selects the asset corresponding to the maximum component of the price-trend vector $\boldsymbol{\delta}$ as the initial portfolio vector $\hat{\boldsymbol{b}}_0$. This choice allows the first term $-\boldsymbol{b}^\intercal\boldsymbol{\delta}$ of the loss function (12) to start with a small value, facilitating accelerated convergence.

6.7. Performance comparison among algorithms

This section presents a comprehensive performance comparison of the proposed ASCM, ASCM$_\alpha$, and ASCM$_\beta$ algorithms with eight prominent algorithms in the field of online portfolio selection. Through this rigorous evaluation, our objective is to gain valuable insights into the respective strengths and weaknesses of each algorithm.

To establish a benchmark, we initiate by examining two widely recognized algorithms: the uniform buy-and-hold (BAH) and the best constant rebalanced portfolio (BCRP) (Cover, 1991; Li & Hoi, 2014). These algorithms' portfolios are defined as per Eqs. (9) and (10), respectively.

Next, we assess three algorithms that fall under the trend-reversion category: online moving average reversion (OLMAR) (Li et al., 2015), Gaussian weighted reversion (GWR) (Cai & Ye, 2019), and trend peak price tracing (TPPT) (Dai et al., 2022). These algorithms leverage trend patterns in the market to make informed investment decisions.

Additionally, we include one ensemble learning algorithm, namely, online expert aggregation (OEA) (He & Yang, 2022), which utilizes ensemble techniques to enhance the robustness of the portfolio selection process. Furthermore, we incorporate a short-term sparse portfolio optimization model that utilizes the l_0-norm formalism (SSPO-l_0) (Wang, Zhang, He, & Cao, 2023). This approach focuses on achieving an optimal sparsity in portfolio construction for short-term trading strategies.

Lastly, we examine an algorithm explicitly developed to facilitate risk control in online portfolio selection, known as short-term portfolio optimization with loss (SPOLC) (Lai et al., 2020). This algorithm incorporates an enhanced mean–variance optimization approach for short-term portfolio selection.

The parameter configurations for these algorithms are outlined below:

- OLMAR: online moving average reversion with the best parameters $\varepsilon = 10$ and $\omega = 5$ as suggested by Li et al. (2015).
- SPOLC: short-term portfolio optimization with loss control with the best parameters $w = 5$ and $\gamma = 0.025$ as suggested by Lai et al. (2020).
- OEA: online expert aggregation with the best parameters $\delta_0 = 0.004$(UP), $\delta = 0.005$(UP), $m = 100$(UP), $S = 500$(UP), $\eta = 0.05$(EG), $\eta = 0$(ONS), $\beta = 1$(ONS) and $\gamma = 0.125$(ONS) as suggested by He and Yang (2022).

Table 6
The top three most frequently occurring numbers of iterations across six datasets.

Method	DJIA			HS300			CRYPTO			TSE			SP500			STOXX50		
	#Iteration	#Period	Proportion	#Iteration	#Period	Proportion	#Iteration	#Period	Proportion	#Iteration	#Period	Proportion	#Iteration	#Period	Proportion	#Iteration	#Period	Proportion
GIM	1	451	89.94%	1	349	84.09%	1	948	86.95%	1	1167	93.09%	1	1124	88.48%	1	3350	86.36%
	2	29	5.71%	2	36	8.55%	2	48	4.38%	2	61	4.85%	2	93	7.29%	2	330	8.49%
	13	6	1.18%	3	11	2.61%	13	10	0.91%	14	11	0.87%	13	15	1.18%	3	53	1.36%
RIM	1	19	3.75%	1	16	3.80%	1	60	1.77%	22	47	3.73%	1	49	3.84%	1	87	2.24%
	34	15	2.96%	31	11	2.61%	20	30	0.89%	24	39	3.10%	33	32	2.51%	35	78	2.01%
	40	13	2.56%	55	10	2.38%	21	29	0.86%	25	38	3.02%	29	29	2.27%	29	76	1.96%
AIM	35	7	1.38%	101	5	1.19%	13	19	1.73%	21	17	1.35%	39	16	1.25%	36	29	0.75%
	52	7	1.38%	50	4	0.95%	20	17	1.55%	28	17	1.35%	33	15	1.18%	42	29	0.75%
	60	7	1.38%	216	4	0.95%	21	17	1.55%	15	16	1.27%	81	15	1.18%	64	25	0.64%

- SSPO-l_0: short-term sparse portfolio optimization model based on l_0-norm with the best parameters $k = 3$, $C = 10$, $W = 5$, $\lambda = 0.1$, $\gamma = 1$ and $\rho = 10$ as suggested by Wang et al. (2023).
- GWR: gaussian weighting reversion with the best parameters $\tau = 2.8$, $\epsilon = 0.005$ and $\delta = 50$ as suggested by Cai and Ye (2019).
- TPPT: trend peak price tracing with the best parameters $\alpha = 0.5$, $w = 5$ and $\varepsilon = 100$ as suggested by Dai et al. (2022).
- ASCM: our asset subset-constrained minimax algorithm with parameters $\{S_\alpha, S_\beta\}$, $w = 5$, $w_\alpha = 9$, $w_\beta = 5$, $\rho_l^\alpha = 0.15$, $\rho_h^\alpha = 0.45$, $\rho_l^\beta = 0.654$, $\rho_h^\beta = 0.734$, $\lambda = 0.002$ and $\epsilon = 5 \times 10^{-4}$.
- ASCM$_\alpha$: our asset subset-constrained minimax algorithm with parameters $\{S, S_\alpha\}$, $w = 5$, $w_\alpha = 9$, $\rho_l^\alpha = 0.15$, $\rho_h^\alpha = 0.45$, $\lambda = 0.002$, and $\epsilon = 5 \times 10^{-4}$.
- ASCM$_\beta$: our asset subset-constrained minimax algorithm with parameters $\{S, S_\beta\}$, $w = 5$, $w_\beta = 5$, $\rho_l^\beta = 0.654$, $\rho_h^\beta = 0.734$, and $\lambda = 0.002$, and $\epsilon = 5 \times 10^{-4}$.

We assess the performance of the aforementioned algorithms using various metrics, including cumulative wealth (CW), annualized return (AR), Sharpe ratio (SR), maximum drawdown (MDD), and Calmar ratio (CAR). These metrics are commonly employed for comparing the effectiveness of online portfolio selection algorithms (Dai et al., 2022; Huang et al., 2016; Lai et al., 2017, 2020). Table 7 displays the performance metrics for all algorithms across the six datasets, assuming zero transaction costs. The highest score in each row is highlighted in bold.

6.7.1. Cumulative wealth

The cumulative wealth (CW), as defined in Eq. (7), serves as a fundamental metric for evaluating the long-term performance of an algorithm.

Table 7 presents the CW values for various algorithms, providing clear evidence of the ASCM algorithm's outstanding performance. It ranks first on four out of the six datasets based on this metric, surpassing all other algorithms except for BCRP on the HS300 dataset. Notably, on the CRYPTO dataset, ASCM achieves an impressive CW of 3155.73, which is more than three times higher than BCRP's CW of 936.13. Similarly, on the TSE dataset, ASCM achieves a CW of 446.22, closely matching the CW of the GWR algorithm. However, the ASCM$_\alpha$ algorithm excels with a remarkable CW of 702.93, ranking first on the TSE dataset. These results clearly demonstrate the capability of the ASCM framework to generate optimal portfolios by leveraging multiple investment strategies.

We present the period-by-period CW of all algorithms, excluding ASCM$_\alpha$ and ASCM$_\beta$, across six datasets in Fig. 8. The CW of the ASCM algorithm is highlighted in red. Notably, ASCM's CW demonstrates robust and consistent growth trends on the DJIA, HS300, TSE, SP500, and STOXX50 datasets, even in the presence of bearish market trends observed in the DJIA and SP500, as indicated by the BAH algorithm.

On the CRYPTO dataset, ASCM's CW exhibits a growth pattern similar to that of BAH, initially increasing until day 600 and then declining. However, ASCM consistently maintains a higher CW than all other algorithms, including BCRP, throughout the entire trading period. This persistent outperformance highlights ASCM's ability to robustly retain strong assets by leveraging the diverse investment strategies S_α and S_β across the full trading horizon.

The CW of ASCM across all datasets highlights its robust risk control capabilities in comparison to the SPOLC algorithm, which follows Markowitz's mean–variance approach to risk control. Notably, the CW of SPOLC underperforms all other algorithms on the CRYPTO dataset, suggesting that in certain emerging markets, actively trading among diverse asset subsets can provide better downside risk control than optimizing mean–variance over short horizons.

6.7.2. Annualized return

An annualized return (AR) evaluates the return on investment, accounting for the compounding effect over a specific number of periods or days. It is calculated as the geometric average of the cumulative wealth earned each year:

$$\mathrm{AR} := (1 + \mathrm{CR})^{\frac{252}{n}} - 1,$$

where $\mathrm{CR} := \frac{\omega_n}{\omega_0} - 1$ represents the cumulative return, and n denotes the number of days.

ASCM emerges as the top performer in terms of AR across four out of the six datasets. On the TSE dataset, ASCM secures the third-place position with an AR of 239.10%, closely trailing the second-place GWR with an AR of 239.52% and the first-place ASCM$_\beta$ with an AR of 271.39%.

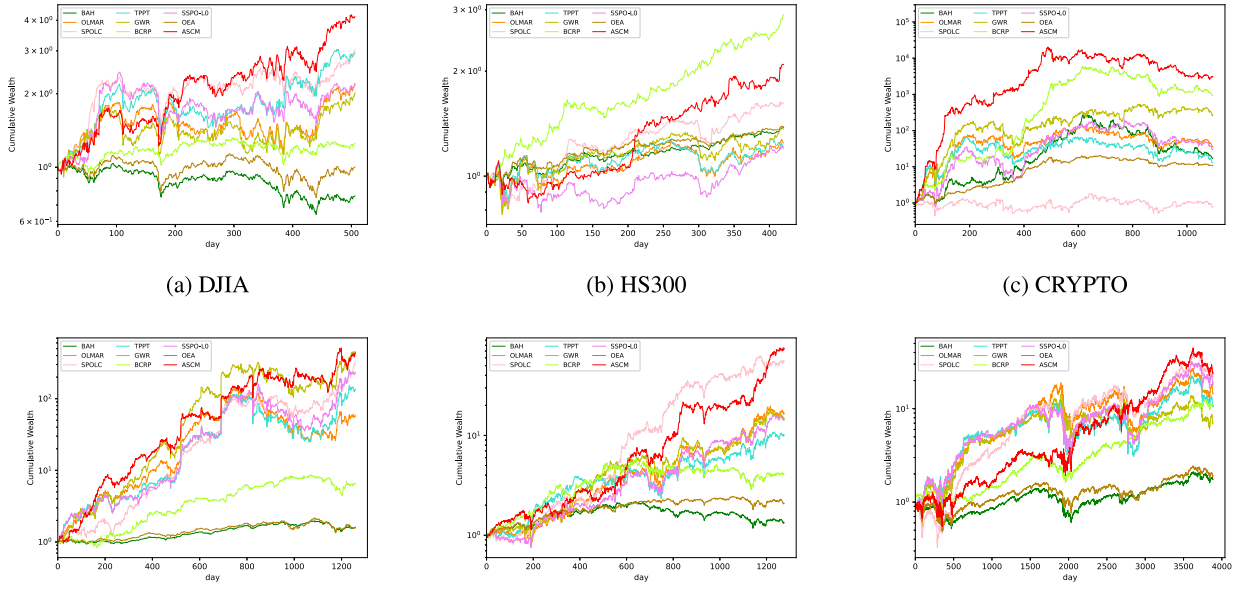
ASCM significantly enhances performance in terms of annualized return (AR) with a remarkable value of 102.5% on the DJIA dataset relative to that of BAH. It is noteworthy that BAH's AR on the DJIA dataset stands out as the only negative value, reaching −12.55%, among all the datasets. This underscores ASCM's exceptional ability to outperform and generate positive returns, even in challenging market conditions like the DJIA.

Despite sharing a similar trend tracking objective, represented as $\boldsymbol{b}^\intercal \boldsymbol{\delta}$ in Eq. (12), ASCM consistently outperforms TPPT and GWR in terms of AR across multiple datasets, including DJIA, HS300, CRYPTO, SP500, and STOXX50. For instance, on the DJIA dataset, ASCM achieves AR values that are 1.45 and 2.46 times higher than those of TPPT and GWR, respectively. This remarkable performance advantage can be attributed to two key factors: (i) ASCM employs diverse investment strategies represented by multiple asset subsets to effectively constrain the feasible region of its trend tracking objective. (ii) ASCM leverages price-trend vectors $\boldsymbol{\delta}$ that encompass both price momentum and price state information.

The AR of SPOLC on the CRYPTO dataset exhibits a striking negative value of −5.35%, despite SPOLC's AR being competitive with TPPT and GWR on other datasets. An intriguing characteristic of the CRYPTO dataset is its power-tailed price distribution, wherein certain assets experience prices exceeding 7000$, while others have prices as low as $8 \times 10^{-5}$$ during specific time intervals. These extreme market conditions pose a significant challenge for SPOLC, which relies on mean–variance based portfolio selection.

Table 7
Algorithm performance comparison.

Dataset	Metric	BAH	BCRP	ASCM	$ASCM_{\alpha}$	$ASCM_{\beta}$	SSPO-l_0	TPPT	SPOLC	GWR	OEA	OLMAR
DJIA	CW	0.76	1.25	**4.14**	3.69	3.31	2.19	2.94	3.03	2.02	1.00	2.20
	AR (%)	−12.55	11.82	**102.50**	91.39	81.41	47.56	70.85	73.60	41.69	0.10	48.00
	SR	−0.55	0.46	**1.36**	1.26	1.13	0.76	1.05	1.16	0.68	0.00	0.77
	MDD (%)	38.29	22.95	**29.79**	30.05	35.22	44.60	35.77	35.52	36.71	33.86	36.85
	CAR	−0.33	0.52	**3.44**	3.04	2.31	1.07	1.98	2.07	1.14	0.00	1.30
HS300	CW	1.38	**2.90**	2.05	1.54	1.66	1.20	1.21	1.62	1.36	1.38	1.25
	AR (%)	21.19	**89.29**	53.61	29.73	35.27	11.7	12.17	33.31	20.47	21.42	14.05
	SR	1.23	**2.8**	1.51	0.92	1.08	0.38	0.41	1.09	0.62	1.21	0.45
	MDD (%)	9.13	11.16	24.40	26.54	25.45	27.33	20.3	19.03	23.85	**9.15**	22.77
	CAR	2.32	8.00	2.20	1.12	1.39	0.43	0.60	1.75	0.86	**2.34**	0.62
CRYPTO	CW	17.34	936.13	**3155.73**	1788.54	759.30	30.62	13.21	0.79	275.88	10.76	37.05
	AR (%)	92.7	382.16	**537.58**	459.55	359.50	119.62	81.01	−5.35	264.07	72.69	129.46
	SR	0.66	1.44	**1.38**	1.30	1.02	0.53	0.43	−0.05	0.92	1.16	0.61
	MDD (%)	94.03	85.22	87.20	79.80	85.16	88.96	55.28	71.92	81.28	**52.60**	79.38
	CAR	0.99	4.48	**6.16**	5.76	4.22	1.34	0.97	−0.07	3.25	1.38	1.63
TSE	CW	1.60	6.60	446.22	278.40	**702.93**	235.67	136.31	334.58	448.95	1.61	59.00
	AR (%)	9.81	45.89	239.10	208.55	**271.39**	198.43	167.45	220.11	239.52	9.96	126.18
	SR	0.72	1.00	1.40	1.29	1.47	1.19	1.09	**1.49**	1.34	0.55	0.90
	MDD (%)	30.02	48.33	54.65	56.40	51.97	77.67	78.58	50.95	67.78	**38.73**	82.01
	CAR	0.33	0.95	4.38	3.70	**5.22**	2.83	2.13	4.32	3.53	0.26	1.54
SP500	CW	1.33	4.05	**75.08**	68.35	28.90	14.43	10.12	55.66	14.85	2.09	16.79
	AR (%)	5.84	31.82	**134.64**	130.33	94.31	69.42	57.95	121.18	70.38	15.64	74.55
	SR	0.23	0.65	**1.63**	1.58	1.26	1.00	0.83	1.60	0.98	0.67	1.04
	MDD (%)	45.8	51.11	39.92	45.88	41.29	45.58	53.32	33.79	48.07	**26.43**	41.17
	CAR	0.13	0.62	3.37	2.84	2.28	1.52	1.09	**3.59**	1.46	0.59	1.81
STOXX50	CW	1.76	10.78	**24.22**	17.31	9.34	19.34	11.13	22.25	6.84	1.92	14.39
	AR (%)	3.75	16.68	**22.96**	20.32	15.60	21.18	16.91	22.29	13.28	4.34	18.88
	SR	0.17	0.68	0.43	0.38	0.28	0.38	0.31	**0.44**	0.25	0.17	0.34
	MDD (%)	57.20	51.37	56.97	60.67	71.26	77.78	81.32	67.60	67.49	**53.17**	74.41
	CAR	0.07	0.32	**0.40**	0.33	0.22	0.27	0.21	0.33	0.20	0.08	0.25



(a) DJIA (b) HS300 (c) CRYPTO

(d) TSE (e) SP500 (f) STOXX50

Fig. 8. Cumulative wealth dynamics.

6.7.3. Sharpe ratio

The Sharpe ratio (SR) quantifies the risk-adjusted return of an investment compared to a risk-free asset. It is calculated as the ratio of the expected excess return to the standard deviation of excess returns:

$$\mathrm{SR} := \frac{\mathbb{E}[R - R_f]}{\sigma[R - R_f]},$$

where R represents the random variable for the daily return of the investment, and R_f represents the daily return of the risk-free asset. In the context of this paper, R_f is set to 0. SR helps evaluate how well an investment compensates for the level of risk taken, and a higher value indicates a more favorable risk-adjusted return.

The ASCM algorithm demonstrates superior performance compared to the other algorithms, except for BCRP, on four out of the six datasets

in terms of SR. This indicates that ASCM is capable of achieving high profits while effectively managing the variance of its daily returns within a reasonable range.

Note that the risk used by the SR metric is measured in the variance of daily returns $\sigma[R - R_f]$, does not differentiate between cases where a sample of the random $R - R_f$ is positive or negative. Consequently, the SR metric has the potential to overestimate the risk when there is actually growth in the daily returns, leading to a potential bias in risk-adjusted evaluation (Gregoriou & Gueyie, 2003).

Moreover, the Sharpe ratio (SR) metric tends to exhibit a preference for mean–variance portfolios over growth-optimal portfolios, primarily due to its lower risk estimation resulting from investments in mean–variance portfolios (Chen, He, & Zhang, 2011). For instance, when examining the STOXX50 dataset in Table 7, even though ASCM outperforms SPOLC in terms of cumulative wealth (CW), annualized return (AR), maximum drawdown (MDD), and Calmar ratio (CAR) metrics (where MDD and CAR are risk-related metrics), ASCM's SR remains at 0.43, which is lower than SPOLC's SR of 0.44.

In the upcoming sections,we will introduce and discuss these risk-related measures in detail. Specifically, in Section 6.7.6, we will propose two novel metrics for evaluating risk-adjusted performance in the context of online portfolio selection.

6.7.4. Maximum drawdown

The maximum drawdown (MDD) is a metric that measures the largest potential percentage decline in the value of a portfolio over a specific period. It serves as an indicator of the potential downside risk within that time frame. Mathematically, it is defined as follows:

$$\mathrm{MDD} := \max_{t\in[1,n]} \max_{\tau\in[1,t]} \frac{\omega_\tau - \omega_t}{\omega_\tau}.$$

where ω_t represents the cumulative wealth at time t, and n denotes the total number of periods.

ASCM achieves the highest rank in MDD performance on the DJIA dataset and secures the second position on the STOXX50 dataset. While on other datasets, ASCM's MDDs closely align with those of other high-performing algorithms, such as SPOLC and GWR, although ASCM's MDDs are slightly higher.

In many market situations, aiming a higher final profit often entails accepting a higher MDD. For example, on the CRYPTO dataset, ASCM achieves an impressive AR of 537.58%, but this accomplishment comes at the expense of a relatively elevated MDD of 87.20%.

Conversely, seeking a lower MDD does not necessarily guarantee a higher final profit. For instance, despite the OEA algorithm excels in MDD performance by securing the top position on five out of six datasets, its CW and AR values significantly lag behind those of the leading algorithms. Specifically, on the HS300 dataset, OEA demonstrates a CW of 1.38 and an AR of 21.42%, which are nearly identical to those of BAH.

6.7.5. Calmar ratio

The Calmar ratio (CAR) (Young, 1991) serves as a measure of performance for investment funds. It calculates the ratio between the fund's annualized return (AR) and its maximum drawdown (MDD):

$$\mathrm{CAR} := \frac{\mathrm{AR}}{\mathrm{MDD}}.$$

ASCM demonstrates better performance on the CAR metric compared to the MDD metric, primarily due to the inclusion of the profit-related metric AR in the CAR calculation formula. Specifically, ASCM achieves the highest CARs on three out of six datasets, secures the second position on TSE and SP500, and ranks third on HS300.

An algorithm can achieve a high CAR by effectively minimizing its MDD, considering CAR as a measure of AR relative to the cost incurred by MDD. For instance, on the HS300 dataset, OEA demonstrates an exceptionally low MDD of 9.15%, which is merely 0.38 times ASCM's MDD. Consequently, OEA secures the top position in terms of CAR, despite its AR being only 0.40 times that of ASCM. This finding underscores the significance of MDD optimization in achieving favorable risk-adjusted performance.

6.7.6. Growth probability and shrinkage probability

The risk-adjusted performance metrics, such as SR and CAR, discussed in the previous sections, may be suboptimal for evaluating online portfolio selection, as they do not directly align with the goal of maximizing capital growth as described in Eq. (8). To address this issue, we propose two novel metrics derived from the cumulative wealth defined in Eq. (7):

$$\omega_n = \underbrace{\prod_{\boldsymbol{x}_t^\intercal \boldsymbol{b}_t \geq 1} \boldsymbol{x}_t^\intercal \boldsymbol{b}_t}_{\omega_g} \underbrace{\prod_{\boldsymbol{x}_t^\intercal \boldsymbol{b}_t < 1} \boldsymbol{x}_t^\intercal \boldsymbol{b}_t}_{\frac{1}{\omega_s}}. \tag{24}$$

where ω_g represents cumulative growth when portfolio vectors $\boldsymbol{b}_t$ align with relative prices $\boldsymbol{x}_t$, while ω_s represents cumulative shrinkage when they are misaligned.

The cumulative growth ω_g and shrinkage ω_s, as defined in Eq. (24), are non-decreasing over time period t. To enable a meaningful comparison of risk-adjusted performance using the ROC (receiver operating characteristic) methodology (Abdelmoula, 2015; Mossman, 2013; Tang & Chi, 2005), we normalize the cumulative growth $\omega_{g,i}$ and shrinkage $\omega_{s,i}$ for each algorithm i into growth probability (GP) $p_{g,i}$ and shrinkage probability (SP) $p_{s,i}$, respectively:

$$p_{g,i} = \frac{a_{g,i}}{a_{g,i} + a_{s,i}},\ p_{s,i} = \frac{a_{s,i}}{a_{g,i} + a_{s,i}}, \tag{25}$$

where the relative growth $a_{g,i}$ and and the relative shrinkage $a_{s,i}$ are defined as:

$$a_{g,i} = \omega_{g,i} / \max\{\omega_{g,j}\} \text{ and } a_{s,i} = \omega_{s,i} / \max\{\omega_{s,j}\},$$

respectively.

Unlike the SR and CAR metrics, it is important to note that the relative growth $a_{g,i}$ and shrinkage $a_{s,i}$ values are derived by scaling the cumulative growth $\omega_{g,i}$ and shrinkage $\omega_{s,i}$ values among all algorithms considered in the performance comparison. This scaling process ensures a consistent basis for comparing the performance of different algorithms. Furthermore, it is worth mentioning that the GP value $p_{g,i}$ and the SP value $p_{s,i}$ satisfy the constant sum condition $p_{g,i} + p_{s,i} = 1$ for each algorithm i.

We introduce the notation $\mathcal{A}_i \succeq \mathcal{A}_j$ to indicate that algorithm i is considered superior to algorithm j in terms of their GP and SP values if the following condition is satisfied:

$$\mathcal{A}_i \succeq \mathcal{A}_j \quad \text{if } p_{g,i} \geq p_{g,j} \text{ and } p_{s,i} \leq p_{s,j}.$$

This condition can be simplified to:

$$\mathcal{A}_i \succeq \mathcal{A}_j \quad \text{if } p_{g,i} \geq p_{g,j},$$

since the equation $p_{g,i} + p_{s,i} = 1$ holds for each algorithm i.

By utilizing the definition provided in Eq. (25), we calculate the GP value $p_{g,i}$ and the SP value $p_{s,i}$ respectively, for each algorithm i. The corresponding values are presented in Table 8. It is evident that, apart from the HS300 and TSE datasets, the conclusion $\mathcal{A}_{\mathrm{ASCM}} \succeq \mathcal{A}_i$ holds true for all algorithms i. On the HS300 dataset, ASCM secures the second position, while BCRP takes the lead. Similarly, on the TSE dataset, ASCM ranks second, with ASCM_β claiming the top position.

To illustrate the distinctions between the GP metric and the CW, SR, and CAR metrics, we utilize the SP500 dataset as an example and present the sorted lists of algorithms in descending order based on these four metrics in Table 9. The ranking of the GP metric aligns with that of the CW metric, indicating a similar assessment of algorithm performance. However, there are different opinions for some ranking positions when comparing GP, SR, and CAR metrics. For example, the 1st, 2nd, and 3rd positions may vary depending on the metric considered. The 1st position could be occupied by ASCM if a voting method is employed across the four metrics, or it could be taken by SPOLC if priority is given to the MDD metric over the CW, AR, and SR metrics.

Table 8
Growth and shrinkage probabilities of algorithms.

Dataset	Metric	BAH	BCRP	ASCM	ASCM$_\alpha$	ASCM$_\beta$	SSPO-l_0	TPPT	SPOLC	GWR	OEA	OLMAR
DJIA	GP	0.20	0.29	**0.58**	0.55	0.53	0.42	0.49	0.50	0.40	0.25	0.42
	SP	0.80	0.71	**0.42**	0.45	0.47	0.58	0.51	0.50	0.60	0.75	0.58
HS300	GP	0.47	**0.65**	0.57	0.50	0.52	0.43	0.44	0.51	0.47	0.47	0.44
	SP	0.53	**0.35**	0.43	0.50	0.48	0.57	0.56	0.49	0.53	0.53	0.56
CRYPTO	GP	0.36	0.97	**0.99**	0.98	0.96	0.50	0.30	0.03	0.90	0.26	0.55
	SP	0.64	0.03	**0.01**	0.02	0.04	0.50	0.70	0.97	0.10	0.74	0.45
TSE	GP	0.01	0.02	0.61	0.49	**0.71**	0.45	0.32	0.54	0.61	0.01	0.17
	SP	0.99	0.98	0.39	0.51	**0.29**	0.55	0.68	0.46	0.39	0.99	0.83
SP500	GP	0.08	0.23	**0.84**	0.81	0.64	0.51	0.42	0.80	0.52	0.13	0.55
	SP	0.91	0.77	**0.16**	0.19	0.36	0.49	0.58	0.20	0.48	0.87	0.45
STOXX50	GP	0.14	0.49	**0.68**	0.64	0.49	0.63	0.50	0.67	0.38	0.15	0.56
	SP	0.86	0.51	**0.32**	0.36	0.51	0.37	0.50	0.33	0.62	0.85	0.44

Table 9
Sorted list of algorithms based on different metrics on the SP500 dataset.

Metric	Algorithms ranked in descending order										
CW	ASCM	ASCM$_\alpha$	SPOLC	ASCM$_\beta$	OLMAR	GWR	SSPO-l_0	TPPT	BCRP	OEA	BAH
GP	ASCM	ASCM$_\alpha$	SPOLC	ASCM$_\beta$	OLMAR	GWR	SSPO-l_0	TPPT	BCRP	OEA	BAH
SR	ASCM	SPOLC	ASCM$_\alpha$	ASCM$_\beta$	OLMAR	SSPO-l_0	GWR	TPPT	OEA	BCRP	BAH
CAR	SPOLC	ASCM	ASCM$_\alpha$	ASCM$_\beta$	OLMAR	SSPO-l_0	GWR	TPPT	BCRP	OEA	BAH

The GP and SP metrics can serve as effective indicators to identify the best-performing market for a specific algorithm. For instance, when considering the ASCM algorithm, the market that demonstrates the highest level of performance is CRYPTO, boasting an exceptional GP of 99% when compared to GP values across various markets. Similarly, the SPOLC algorithm exhibits exceptional performance in the SP500 market, with a GP of 80%, surpassing all other GP values across diverse markets.

The GP and SP values of all algorithms can be visualized in a two-dimensional plane, providing an insightful representation. Each data point in Fig. 9 represents the combined performance of an algorithm on the GP and SP metrics. Notably, algorithms positioned at the uppermost and left-most locations signify the top performers in terms of GP and SP metrics, respectively, indicating their superior performance compared to others in the evaluation.

6.7.7. Impact of transaction costs

In this section, we employ the proportional transaction cost model (Lai et al., 2017, 2020) to evaluate the practical applicability of the ASCM algorithm by analyzing the dynamics of its cumulative wealth (CW) under varying transaction costs.

If transaction costs are taken into account, the definition of cumulative wealth in Eq. (7) needs to be modified to $w_t = w_0 \prod_{\tau=1}^{t} \mu_\tau \boldsymbol{x}_\tau^\intercal \boldsymbol{b}_\tau$, where the symbol $\mu_\tau \in (0, 1)$ represents a decay factor at period τ. The determination of the decay factor μ_τ can be accomplished using the method introduced in Gyorfi and Walk (2012) under the self-financing assumption.

We investigate a range of transaction cost values, spanning from 0% to 0.5%, and record the cumulative wealth of all algorithms, as depicted in Fig. 10. Most algorithms exhibit a linear decline in their cumulative wealth as transaction costs increase. For instance, on the DJIA dataset, the cumulative wealth of ASCM experiences a significant decrease, dropping from approximately 4 to nearly 2. Conversely, algorithms like BCRP, BAH, and OEA maintain relatively stable cumulative wealth despite the increase in transaction costs. This behavior can be attributed to their lower turnover values, which reflect a more conservative trading approach.

The ASCM algorithm consistently outperforms most algorithms, consistently achieving the highest cumulative wealth across the majority of datasets, particularly when transaction costs are below 0.1%. The

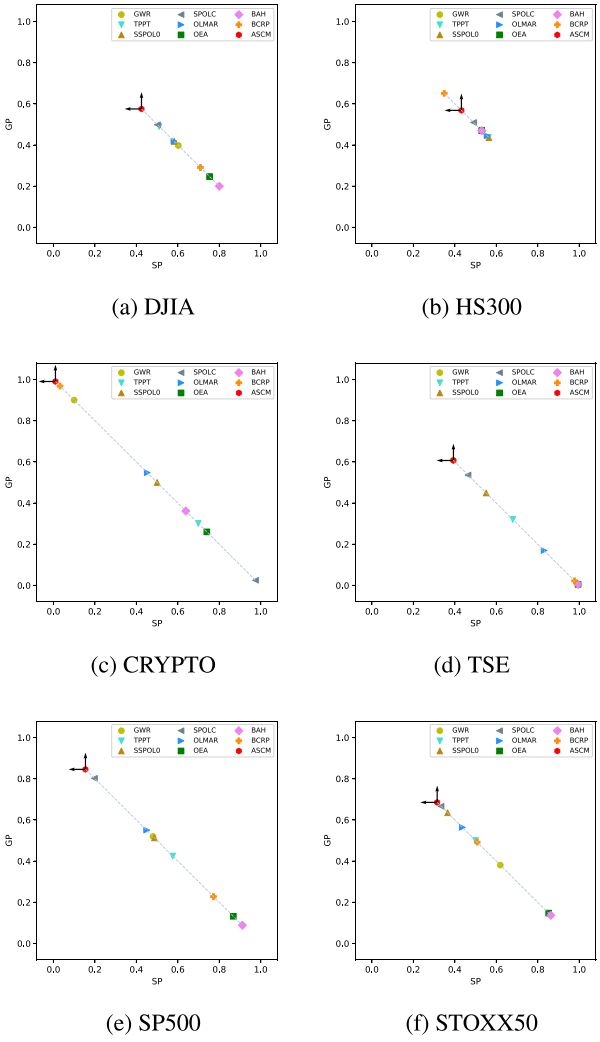
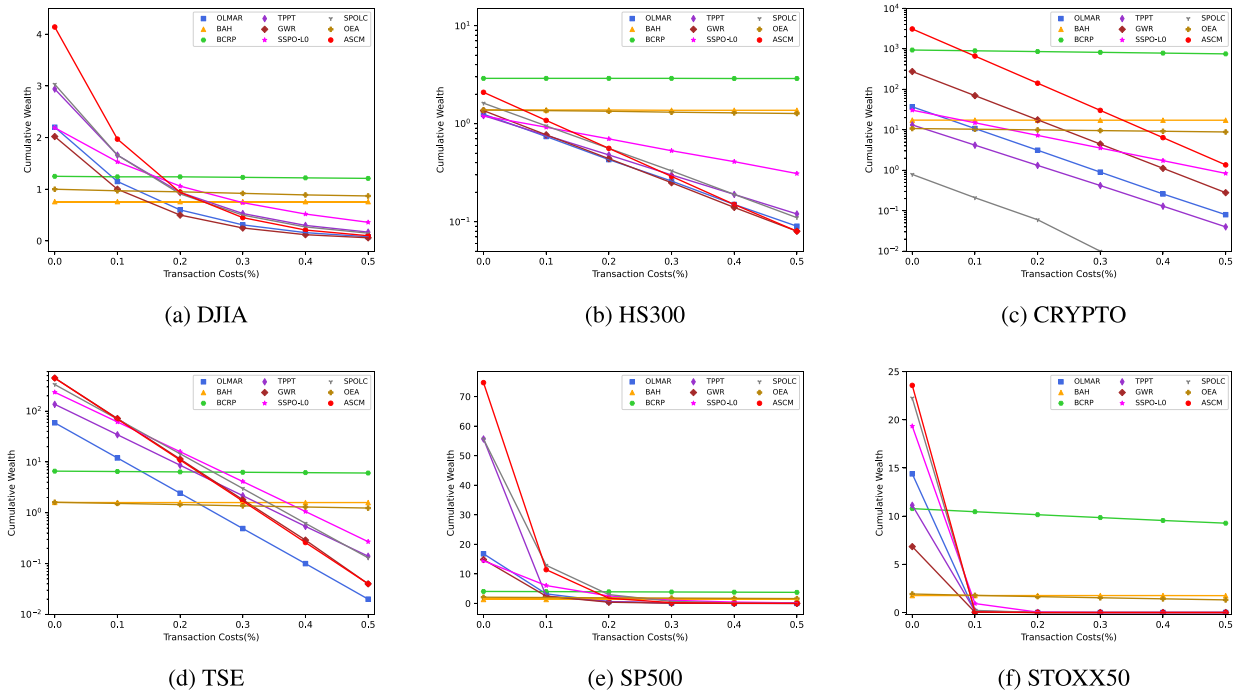


Fig. 9. Visualizing the probabilities of growth and shrinkage for algorithms.



(a) DJIA (b) HS300 (c) CRYPTO

(d) TSE (e) SP500 (f) STOXX50

Fig. 10. Impact of transaction costs on cumulative wealth.

ASCM algorithm exhibits a relatively higher decline rate in cumulative wealth due to its dynamic switching among multiple asset subsets, distinguishing it from algorithms like OEA and TPPT.

It is important to acknowledge the challenge of accurately estimating transaction costs, as they encompass fixed and variable costs along with visible and hidden components. Fixed costs are independent of market prices and cannot be avoided, while variable costs depend on market prices and can be managed through implementation strategies. Visible costs, such as commissions and taxes, have observable cost structures within the market, whereas non-transparent costs, like market impact, are not easily discernible. Studies (Berkowitz, Logue, & Noser, 1988; Lesmond, Ogden, & Trzcinka, 1999; Li et al., 2015) have shown that transaction costs in real financial markets are generally much lower than 0.1%. Consequently, the ASCM algorithm proves to be suitable for practical transaction environments.

7. Conclusion

This paper introduces the asset subset-constrained minimax (ASCM) optimization framework, which presents an innovative solution to the critical concerns of online portfolio selection: diversity, sparsity, and risk control. By effectively addressing these concerns simultaneously, ASCM makes a significant contribution to the field of online portfolio selection by offering a holistic and integrated approach.

The ASCM optimization model presented in this paper empowers effective risk control across multiple investment strategies represented by diverse asset subsets. By Leveraging a projected subgradient optimization method, we successfully solve the ASCM model, attaining a remarkable convergence speed with an approximate $O(1)$ time complexity, aided by our novel initialization scheme.

Experiments confirm ASCM's superior performance over state-of-the-art approaches across six real-world datasets from diverse financial markets. For instance, on the CRYPTO dataset, ASCM significantly outperforms the second-ranked algorithm with a remarkable cumulative wealth factor of 11.4. Across six datasets, ASCM consistently demonstrates robust balance between returns and risks in terms of Sharpe ratio, Calmar ratio, growth probability and shrinkage probability compared to other evaluated algorithms.

The extensible nature of the ASCM optimization framework creates promising opportunities to design novel trading algorithms that can effectively navigate complex trading environments. As a demonstration of its extensibility, we propose three algorithms, namely ASCM, ASCM_{α}, and ASCM_{β}, each leveraging distinct asset subsets. Remarkably, the combination of these three algorithms achieves top-ranked performance across all six datasets.

Future research endeavors can delve into pioneering methodologies to construct specialized asset subsets and design novel price-trend vectors, consequently augmenting the performance and adaptability of the ASCM framework. This opens up exciting opportunities for advancing the field and pushing the boundaries of online portfolio selection.

CRediT authorship contribution statement

Jianfei Yin: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft. **Anyang Zhong:** Data curation, Methodology, Investigation, Software, Visualization, Writing – review & editing. **Xiaomian Xiao:** Data curation, Software. **Ruili Wang:** Supervision, Methodology, Writing – review & editing. **Joshua Zhexue Huang:** Data curation, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets and source codes are available at https://github.com/xiterator/ASCM.

Acknowledgments

The authors express gratitude to the reviewers for their valuable comments and suggestions. The authors also acknowledge the partial support from the Key Basic Research Foundation of Shenzhen (JCYJ20220818100205012) and the Fundamental Research Funds for the Central Universities (Grant No. 3132023523).

References

Abdelmoula, A. K. (2015). Bank credit risk analysis with k-nearest-neighbor classifier: Case of Tunisian banks. *Accounting and Management Information Systems*, *14*(1), 79–106.

Agarwal, A., Hazan, E., Kale, S., & Schapire, R. E. (2006). Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd international conference on machine learning* (pp. 9–16).

Alameer, A., & Al Shehri, K. (2022). Conditional value-at-risk for quantitative trading: A direct reinforcement learning approach. In *2022 IEEE conference on control technology and applications* (pp. 1208–1213). IEEE.

Anis, H. T., & Kwon, R. H. (2022). Cardinality-constrained risk parity portfolios. *European Journal of Operational Research*, *302*(1), 392–402.

Aquino, L. D. G., Sornette, D., & Strub, M. S. (2023). Portfolio selection with exploration of new investment assets. *European Journal of Operational Research*, *310*(2), 773–792.

Bai, Y., Yin, J., Ju, S., Chen, Z., & Huang, J. Z. (2020). Long and short term risk control for online portfolio selection. In *International conference on knowledge science, engineering and management* (pp. 472–480). Springer.

Baron, M., Brogaard, J., Hagströmer, B., & Kirilenko, A. (2019). Risk and return in high-frequency trading. *Journal of Financial and Quantitative Analysis*, *54*(3), 993–1024.

Berkowitz, S. A., Logue, D. E., & Noser, E. A., Jr. (1988). The total cost of transactions on the NYSE. *The Journal of Finance*, *43*(1), 97–112.

Bhatt, A., Ryu, J. J., & Kim, Y. H. (2023). On universal portfolios with continuous side information. In *International conference on artificial intelligence and statistics* (pp. 4147–4163). PMLR.

Borodin, A., El-Yaniv, R., & Gogan, V. (2003). Can we learn to beat the best stock. *Advances in Neural Information Processing Systems*, *16*.

Brito, I. (2023). A portfolio stock selection model based on expected utility, entropy and variance. *Expert Systems with Applications*, *213*, Article 118896.

Brodie, J., Daubechies, I., De Mol, C., Giannone, D., & Loris, I. (2009). Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, *106*(30), 12267–12272.

Cai, X., & Ye, Z. (2019). Gaussian weighting reversion strategy for accurate online portfolio selection. *IEEE Transactions on Signal Processing*, *67*(21), 5558–5570.

Carleo, A., Cesarone, F., Gheno, A., & Ricci, J. M. (2017). Approximating exact expected utility via portfolio efficient frontiers. *Decisions in Economics and Finance*, *40*(1), 115–143.

Chen, M. Y., Chen, C. T., & Huang, S. H. (2023). Knowledge distillation for portfolio management using multi-agent reinforcement learning. *Advanced Engineering Informatics*, *57*, Article 102096.

Chen, L., He, S., & Zhang, S. (2011). When all risk-adjusted performance measures are the same: In praise of the Sharpe ratio. *Quantitative Finance*, *11*(10), 1439–1447.

Chen, Y., Wiesel, A., & Hero, A. O. (2011). Robust shrinkage estimation of high-dimensional covariance matrices. *IEEE Transactions on Signal Processing*, *59*(9), 4097–4107.

Chen, Y., & Zhou, A. (2022). Multiobjective portfolio optimization via Pareto front evolution. *Complex & Intelligent Systems*, *8*(5), 4301–4317.

Chiu, W. Y. (2021). Safety-first portfolio selection. *Mathematics and Financial Economics*, *15*(3), 657–674.

Cover, T. M. (1991). Universal portfolios. *Mathematical Finance*, *1*(1), 1–29.

Cover, T. M., & Ordentlich, E. (1996). Universal portfolios with side information. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *42*(2), 348–363.

Cui, T., Du, N., Yang, X., & Ding, S. (2024). Multi-period portfolio optimization using a deep reinforcement learning hyper-heuristic approach. *Technological Forecasting and Social Change*, *198*, Article 122944.

Dai, Z., & Kang, J. (2022). Some new efficient mean–variance portfolio selection models. *International Journal of Finance & Economics*, *27*(4), 4784–4796.

Dai, H. L., Liang, C. X., Dai, H. M., Huang, C. Y., & Adnan, R. M. (2022). An online portfolio strategy based on trend promote price tracing ensemble learning algorithm. *Knowledge-Based Systems*, *239*, Article 107957.

Daluiso, R., Pinciroli, M., Trapletti, M., & Vittori, E. (2023). Cva hedging with reinforcement learning. In *Proceedings of the fourth ACM international conference on AI in finance* (pp. 261–269).

Fereydooni, A., Barak, S., & Sajadi, S. M. A. (2024). A novel online portfolio selection approach based on pattern matching and ESG factors. *Omega*, *123*, Article 102975.

Fulga, C., Dedu, S., & Şerban, F. (2009). Portfolio optimization with prior stock selection. *Economic Computation and Economic Cybernetics Studies and Research*, *43*(4), 157–172.

Gregoriou, G. N., & Gueyie, J. P. (2003). Risk-adjusted performance of funds of hedge funds using a modified sharpe ratio. *The Journal of Wealth Management*, *6*(3), 77–83.

Gubu, L., Rosadi, D., et al. (2021). A new approach for robust mean-variance portfolio selection using trimmed k-means clustering. *Industrial Engineering & Management Systems*, *20*(4), 782–794.

Gunjan, A., & Bhattacharyya, S. (2023). A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, *56*(5), 3847–3886.

Gyorfi, L., & Walk, H. (2012). Empirical portfolio selection strategies with proportional transaction costs. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *58*(10), 6320–6331.

Han, J., & Ge, Z. (2020). Effect of dimensionality reduction on stock selection with cluster analysis in different market situations. *Expert Systems with Applications*, *147*, Article 113226.

He, J., & Yang, X. (2022). Universal portfolio selection strategy by aggregating online expert advice. *Optimization and Engineering*, *23*, 85–109.

Helmbold, D. P., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1998). On-line portfolio selection using multiplicative updates. *Mathematical Finance*, *8*(4), 325–347.

Hosseinzadeh, M. M., Ortobelli Lozza, S., Hosseinzadeh Lotfi, F., & Moriggia, V. (2023). Portfolio optimization with asset preselection using data envelopment analysis. *Central European Journal of Operations Research*, *31*(1), 287–310.

Huang, D., Zhou, J., Li, B., Hoi, S. C., & Zhou, S. (2016). Robust median reversion strategy for online portfolio selection. *IEEE Transactions on Knowledge and Data Engineering*, *28*(9), 2480–2493.

Jalota, H., Mandal, P. K., Thakur, M., & Mittal, G. (2023). A novel approach to incorporate investor's preference in fuzzy multi-objective portfolio selection problem using credibility measure. *Expert Systems with Applications*, *212*, Article 118583.

Ji, G., Yu, J., Hu, K., Xie, J., & Ji, X. (2022). An adaptive feature selection schema using improved technical indicators for predicting stock price movements. *Expert Systems with Applications*, *200*, Article 116941.

Kobayashi, K., Takano, Y., & Nakata, K. (2021). Bilevel cutting-plane algorithm for cardinality-constrained mean-CVaR portfolio optimization. *Journal of Global Optimization*, *81*(2), 493–528.

Lai, Z. R., Dai, D. Q., Ren, C. X., & Huang, K. K. (2017). A peak price tracking-based learning system for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(7), 2823–2832.

Lai, Z. R., Dai, D. Q., Ren, C. X., & Huang, K. K. (2018). Radial basis functions with adaptive input and composite trend representation for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(12), 6214–6226.

Lai, Z. R., Li, C., Wu, X., Guan, Q., & Fang, L. (2022). Multitrend conditional value at risk for portfolio optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 1545–1558.

Lai, Z. R., Tan, L., Wu, X., & Fang, L. (2020). Loss control with rank-one covariance estimate for short-term portfolio optimization. *Journal of Machine Learning Research*, *21*(97), 1–37.

Lai, Z. R., & Yang, H. (2022). A survey on gaps between mean-variance approach and exponential growth rate approach for portfolio optimization. *ACM Computing Surveys*, *55*(2), 1–36.

Ledoit, O., & Wolf, M. (2022). The power of (non-) linear shrinking: A review and guide to covariance matrix estimation. *Journal of Financial Econometrics*, *20*(1), 187–218.

Leippold, M., Trojani, F., & Vanini, P. (2011). Multiperiod mean-variance efficient portfolios with endogenous liabilities. *Quantitative Finance*, *11*(10), 1535–1546.

Lesmond, D. A., Ogden, J. P., & Trzcinka, C. A. (1999). A new estimate of transaction costs. *The Review of Financial Studies*, *12*(5), 1113–1141.

Leung, M. F., & Wang, J. (2020). Minimax and biobjective portfolio selection based on collaborative neurodynamic optimization. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(7), 2825–2836.

Lezmi, E., Roncalli, T., & Xu, J. (2022). Multi-period portfolio optimization. Available at SSRN 4078043.

Li, T., Chen, K., Feng, Y., & Ying, Z. (2017). Binary switch portfolio. *Quantitative Finance*, *17*(5), 763–780.

Li, B., & Hoi, S. C. (2014). Online portfolio selection: A survey. *ACM Computing Surveys*, *46*(3), 1–36.

Li, B., Hoi, S. C., Sahoo, D., & Liu, Z. Y. (2015). Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, *222*, 104–123.

Li, Y., & Mi, H. (2021). Portfolio optimization under safety first expected utility with nonlinear probability distortion. *Chaos, Solitons & Fractals*, *147*, Article 110917.

Li, Q., Qin, Z., & Yan, Y. (2022). Uncertain random portfolio optimization model with tail value-at-risk. *Soft Computing*, *26*(18), 9385–9394.

Li, X. P., Shi, Z. L., Leung, C. S., & So, H. C. (2022). Sparse index tracking with K-sparsity or ε-deviation constraint via l_0-norm minimization. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(12), 10930–10943.

Li, X., Uysal, A. S., & Mulvey, J. M. (2022). Multi-period portfolio optimization using model predictive control with mean-variance and risk parity frameworks. *European Journal of Operational Research*, *299*(3), 1158–1176.

Li, C., Wu, Y., Lu, Z., Wang, J., & Hu, Y. (2020). A multiperiod multiobjective portfolio selection model with fuzzy random returns for large scale securities data. *IEEE Transactions on Fuzzy Systems*, *29*(1), 59–74.

Li, Q., Zhang, W., Wang, G., & Bai, Y. (2023). Non-convex regularization and accelerated gradient algorithm for sparse portfolio selection. *Optimization Methods & Software*, *38*(2), 434–456.

Lin, H., Zhang, Y., & Yang, X. (2024). Online portfolio selection of integrating expert strategies based on mean reversion and trading volume. *Expert Systems with Applications*, *238*, Article 121472.

Livan, G., Inoue, J. i., & Scalas, E. (2012). On the non-stationarity of financial time series: impact on optimal portfolio selection. *Journal of Statistical Mechanics: Theory and Experiment*, *2012*(07), Article P07025.

Lorenzo, L., & Arroyo, J. (2023). Online risk-based portfolio allocation on subsets of crypto assets applying a prototype-based clustering algorithm. *Financial Innovation*, *9*(1), Article 25.

Luan, F., Zhang, W., & Liu, Y. (2022). Robust international portfolio optimization with worst-case mean-CVaR. *European Journal of Operational Research*, *303*(2), 877–890.

Luo, H., Wei, C.-Y., & Zheng, K. (2018). Efficient online portfolio with logarithmic regret. *Advances in Neural Information Processing Systems*, *31*.

Luo, Z., Yu, X., Xiu, N., & Wang, X. (2022). Closed-form solutions for short-term sparse portfolio optimization. *Optimization*, *71*(7), 1937–1953.

MacLean, L. C., Thorp, E. O., & Ziemba, W. T. (2010). Long-term capital growth: the good and bad properties of the Kelly and fractional Kelly capital growth criteria. *Quantitative Finance*, *10*(7), 681–687.

Mazraeh, N. B., Daneshvar, A., Roodposhti, F. R., et al. (2022). Stock portfolio optimization using a combined approach of multi objective grey wolf optimizer and machine learning preselection methods. *Computational Intelligence and Neuroscience*, *2022*.

Mhammedi, Z., & Rakhlin, A. (2022). Damped online Newton step for portfolio selection. In *Conference on learning theory* (pp. 5561–5595). PMLR.

Mossman, D. (2013). Evaluating risk assessments using receiver operating characteristic analysis: Rationale, advantages, insights, and limitations. *Behavioral Sciences & the Law*, *31*(1), 23–39.

Niu, H., Li, S., & Li, J. (2022). MetaTrader: An reinforcement learning approach integrating diverse policies for portfolio optimization. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 1573–1583).

Petukhina, A., Klochkov, Y., Härdle, W. K., & Zhivotovskiy, N. (2023). Robustifying markowitz. *Journal of Econometrics*, *239*(2), Article 105387.

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J. F., Breazeal, C., et al. (2019). Machine behaviour. *Nature*, *568*(7753), 477–486.

Sengupta, J. K. (1989). Portfolio decisions as games. *International Journal of Systems Science*, *20*(8), 1323–1334.

Shavandi, A., & Khedmati, M. (2022). A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications*, *208*, Article 118124.

Shen, W., & Wang, J. (2017). Portfolio selection via subset resampling. In *Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1* (pp. 1517–1523).

Shi, Z. L., Li, X. P., Leung, C. S., & So, H. C. (2022). Cardinality constrained portfolio optimization via alternating direction method of multipliers. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 2901–2909.

Shynkevich, Y., McGinnity, T. M., Coleman, S. A., Belatreche, A., & Li, Y. (2017). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, *264*, 71–88.

Soleymani, F., & Vasighi, M. (2022). Efficient portfolio construction by means of CVaR and k-means++ clustering analysis: Evidence from the NYSE. *International Journal of Finance & Economics*, *27*(3), 3679–3693.

Song, Y., Suganthan, P. N., Pedrycz, W., Ou, J., He, Y., Chen, Y., et al. (2023). Ensemble reinforcement learning: A survey. *Applied Soft Computing*, *149*, Article 110975.

Spelta, A., Pecora, N., & Pagnottoni, P. (2022). Chaos based portfolio selection: A nonlinear dynamics approach. *Expert Systems with Applications*, *188*, Article 116055.

Steinbach, M. C. (2001). Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM Review*, *43*(1), 31–85.

Tang, T. C., & Chi, L. C. (2005). Predicting multilateral trade credit risks: comparisons of Logit and Fuzzy Logic models using ROC curve analysis. *Expert Systems with Applications*, *28*(3), 547–556.

Treynor, J. (1999). Zero sum. *Financial Analysts Journal*, *55*(1), 8–12.

Tsantekidis, A., Passalis, N., & Tefas, A. (2021). Diversity-driven knowledge distillation for financial trading using deep reinforcement learning. *Neural Networks*, *140*, 193–202.

Wang, W., Li, W., Zhang, N., & Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, *143*, Article 113042.

Wang, H., Zhang, W., He, Y., & Cao, W. (2023). l0-norm based short-term sparse portfolio optimization algorithm based on alternating direction method of multipliers. *Signal Processing*, *208*, Article 108957.

Wei, J., Liu, X., & Fan, W. (2022). Dynamic sparse portfolio rebalancing model: A perspective of investors' behavior-related decisions. *Knowledge-Based Systems*, *251*, Article 109224.

Wei, J., Yang, Y., Jiang, M., & Liu, J. (2021). Dynamic multi-period sparse portfolio selection model with asymmetric investors' sentiments. *Expert Systems with Applications*, *177*, Article 114945.

Wu, Z., Xie, G., Ge, Z., & De Simone, V. (2024). Nonconvex multi-period mean-variance portfolio optimization. *Annals of Operations Research*, *332*, 617–644.

Yang, X., He, J., Lin, H., & Zhang, Y. (2020). Boosting exponential gradient strategy for online portfolio selection: an aggregating experts' advice method. *Computational Economics*, *55*(1), 231–251.

Yang, X., He, J., Xian, J., Lin, H., & Zhang, Y. (2020). Aggregating expert advice strategy for online portfolio selection with side information. *Soft Computing*, *24*, 2067–2081.

Yang, H., Liu, X. Y., Zhong, S., & Walid, A. (2020). Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance* (pp. 1–8).

Yin, X. C., Huang, K., Yang, C., & Hao, H. W. (2014). Convex ensemble learning with sparsity and diversity. *Information Fusion*, *20*, 49–59.

Yin, J., Wang, R., Guo, Y., Bai, Y., Ju, S., Liu, W., et al. (2021). Wealth flow model: Online portfolio selection based on learning wealth flow matrices. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *16*(2), 1–27.

Yin, J., Wang, R., Ju, S., Bai, Y., & Huang, J. Z. (2020). An asymptotic statistical learning algorithm for prediction of key trading events. *IEEE Intelligent Systems*, *35*(2), 25–35.

Young, T. W. (1991). Calmar ratio: A smoother tool. *Futures*, *20*(1), Article 40.

Zheng, Y., Hospedales, T. M., & Yang, Y. (2020). Diversity and sparsity: A new perspective on index tracking. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 1768–1772). IEEE.

Zheng, Y., & Zheng, J. (2022). A novel portfolio optimization model via combining multi-objective optimization and multi-attribute decision making. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *52*(5), 5684–5695.