

Rule Constrained Reinforcement Learning for Portfolio Optimization in the OMX Baltics Index

Ilja Jurcenko^{1,*}, Igoris Belovas^{1,†}

¹*Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4, Vilnius, LT-08412, Lithuania*

Abstract

This research addresses the problem of portfolio optimization using reinforcement learning extended with rule-based constraints. The deep deterministic policy gradient is updated with a financial ruleset for transaction, diversification, and risk control. The rules are embedded into the reward function to indicate violations and penalties. The methodology is applied to companies in the OMX Baltic Index, representing an emerging market setting and providing a novel empirical context for the study. The results have potential for practical insight to portfolio managers and fill the gap of RL application with constraints in Baltics market.

Keywords

Portfolio Optimization, Reinforcement Learning, OMX Baltics, Rule Constrained, Financial Engineering

1. Introduction

Financial assets, such as stocks, bonds, cash, and commodities, as well as other various financial instruments owned by an investor are put together to form a portfolio to grow wealth while managing risk. For example, to achieve investors' financial goals, portfolio management involves selecting and optimizing asset weights according to the level of risk an investor is willing to accept. In other words, the portfolio optimization problem involves optimization in a constrained environment, where investor risk tolerance, market exposure, and budget must be addressed. Additionally, declarative explicit rules are formulated by regulators to define a constrain space - maximum asset weights, maximum allowed turnover, sector exposure limits or Value-at-Risk thresholds.

A modern approach to the portfolio optimization problem was introduced by Markowitz [1]. The presented mean-variance optimization holds static assumptions - weight calculations based on historical data distribution and the normality of asset returns [1]. Due to the higher complexity of financial markets, these assumptions fail to generalize. Thus, the portfolio manager must make decisions and perform calculations in a dynamic, uncertain environment with reduced accuracy. The primary approach of obtaining portfolio weights based on data distribution is challenged by dynamic nature of the market. That is why nowadays machine learning approaches are a popular application for portfolio management, because allocation rules can be derived from patterns in the data instead of data distribution assumptions. Furthermore, financial data is not labeled for a supervised learning approach. Thus, the portfolio optimization problem is currently solved using reinforcement learning approaches.

The primary focus of the RL approach is risk-return optimization, which is not the case in practice, as portfolio management is rarely unconstrained. Explicit allocation rules must be addressed during RL optimization to obtain proper results. The research question - how RL can be extended with declarative investment rules so that the agent optimizes the portfolio to meet profitability targets while remaining compliant with predefined constraints within OMX Baltics Index?

The novelty and originality of the work can be formulated as an application and evaluation of a declarative rule-constrained RL framework for portfolio optimization in OMX Baltic market. The

RuleML+RR'26: Companion Proceedings of the 10th International Joint Conference on Rules and Reasoning, August 24–26, 2026, Vilnius, Lithuania

*Corresponding author.

† These authors contributed equally.

✉ ilja.jurcenko@mif.stud.vu.lt (I. Jurcenko); igoris.belovas@mif.vu.lt (I. Belovas)

ORCID 0009-0003-6090-3973 (I. Jurcenko); 0000-0002-0478-1102 (I. Belovas)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

motivation is to apply constrained optimization in emerging markets, such as the Baltic market, since most of the work focuses on developed markets. The rules for emerging and developed markets can differ due to differences in volatility and other properties.

2. Related Works

Enkhsaikhan and Jo [2] propose a risk-constrained variant of the RL methodology by applying the augmented Lagrangian multiplier. The agent's primary goal is to maximize portfolio return, while the Lagrangian multiplier penalizes violations of risk measures, such as Value-at-Risk (VaR). The authors formulate the problem as a constrained Markov decision process, in which the reward is not simply penalized but is constrained by expected future discounted risk. The RL agent is exploring profitable portfolio weights, but the selections must keep the estimated VaR below a certain threshold. In other words, if the agent keeps selecting bad portfolio weights, the reward function will motivate to explore safer options. The approach showed promising results, with no constraint violations during testing and significant benchmark outperformance. In this paper, the rule-based approach of risk control is integrated into the reward function as an augmented Lagrangian multiplier.

The portfolio allocation rules can be formulated using reward function only. Jiang et al. [3] use twin delayed deterministic policy gradient (TD3) to find optimal portfolio weights with a modified reward function. The reward function is extended to have transaction costs and risk rule formulation. The risk control is formulated using Markowitz's mean-variance approach, with portfolio variance multiplied by the investor's risk aversion coefficient. It is a softer constrain approach compared to the lagrangian multiplier. Additionally, the rule is enriched with transaction costs control to prevent costly and frequent portfolio turnovers. The ML rule is formulated for risk adjusted return maximization instead of zero risk violations.

Additionally, minimum sector allocation rules are formulated. Winkel et al. [4] describe an asset allocation percentage rule for the RL domain using a simplex method. The authors implemented the simplex method to formulate a rule for the minimum percentage investments in the "green" and "fossil fuel" sectors. The approach is to decompose the action space into four sub-action spaces. These spaces are combined to finally form the final portfolio allocation. In this way, the rule-based approach of RL guarantees to converge with the set ruleset for sector allocation that satisfy proportions set by a financial manager.

Gollart and Okhrin [5] state that the DDP3 RL algorithm is well-suited for formulating action constraints. The model restricts allocation to risky assets, and weight bounds control the algorithm to prevent concentrated wealth in risky assets. Another example of similar approach is provided by Beyhan et al. [6]. The RL algorithm DDPG is applied to optimize a portfolio in emerging-market settings. The primary novelty is its application to emerging-market data from the Istanbul BIST30 index, whereas other works primarily focus on developed markets such as the USA. The rule control focuses on diversification, as the reward function is extended to include portfolio-level entropy to mitigate and penalize concentration.

Furthermore, Li et al. [7] addresses the problem of integrating RL-based approaches with rule-based constraints. The authors add not only the optimization task but also the asset selection task to the algorithm. In other words, the cardinality constraint is added as a maximum number selection of assets from the universe, and the bounding constraint, where each asset must have a weight between a lower bound and an upper bound. To address the constraints, the paper proposes a safety layer approach. The layer that uses the Mean-Variance model is responsible for mapping the wrong actions that violate the constraints into a safe, feasible space. Moreover, it formulates the problem as a constrained Markov Decision Process and strictly penalizes violations of the constraints using a logarithmic barrier function. The rule-based control is achieved through penalty and special-layer corrections.

Li et al. [8] aim to address the problem of conflicting constraint rules by introducing a multi-agent approach. Compared with traditional approaches that focus on modifying reward functions, the multi-agent approach employs a dual RL agent architecture. The primary agent's goal is profit maximization,

which passes suggested weights as input to a second agent called the Heuristic-based agent. The second agent's aim is on constraint satisfaction and corrections. This architecture can support a set of rules: risk, holdings, cardinality, turnover, sector or class, and capitalization. The authors report significant results relative to benchmarks, underscoring the importance of their findings.

To conclude, the most recent works of Enkhsaikhan and Jo [2] and Winkel et al. [4] introduce optimization methods such as the Lagrangian multiplier for risk constraints and simplex for constrained percentage allocations. A more straightforward approach is to extend the reward function to penalize constrain violations, which is introduced in the works of Jiang et al. [3], Gollart and Okhrin [5], Beyhan et al. [6] with transaction costs, diversification and risk formulation. Another example is a special mapping and correction layer approach Li et al. [7] for violating actions. In contrast with other methods, Li et al. [8] present a multi-agent approach. These approaches aim to integrate rule-based corrections into RL training, but most of the work focuses on developed markets, where the rules may differ from those in emerging markets. Thus, it is a research gap that this paper is trying to address for the Baltic market.

3. Research Methodology

In a Markovian framework with state transitions and associated probabilities, RL can be combined with hidden-state estimation to identify latent market regimes and optimize investment decisions accordingly.

Algorithm 1 Generic Rule-Constrained RL algorithm for Portfolio Optimization

```

1: Input: Historical market data of OMX Baltics, portfolio rule set  $\mathcal{R}$ , RL agent as DDPG architecture, training horizon  $T$ 
2: Output: Rule compliant asset allocation policy
3: Initialize the RL agent and portfolio environment
4: Define the state space, action space, reward function, and rules of constraints
5: Initialize the portfolio weights
6: for each training episode do
7:   Observe the current market state  $s_t$ 
8:   for each trading time step  $t = 1, \dots, T$  do
9:     Select a action  $a_t$  under current policy
10:    Check whether  $a_t$  satisfies constrain rules  $\mathcal{R}$ 
11:    if  $a_t$  violates any rule then
12:      Compute the constraint violation
13:      Penalize, reject, or modify the action by mapping layer
14:    end if
15:    Execute the action and fill the orders
16:    Observe portfolio return, Sharpe ratio, transaction cost, and next state  $s_{t+1}$ 
17:    Compute the reward function using return, cost, and possible rule-violation terms
18:    Store the transition for learning in the replay buffer
19:    Perform parameter updates for the policy networks
20:   end for
21: end for
22: return Trained rule-constrained portfolio allocation policy

```

Machine learning can be categorized by several learning paradigms, such as supervised, unsupervised, and reinforcement learning. The supervised approach trains a system with labeled data, while the unsupervised approach automatically finds patterns in data, primarily used in clustering algorithms. The RL approach is different – it requires an agent to operate in the environment using a reward function and reward signals [9]. The problem of portfolio rebalancing from an RL perspective can be viewed as an agent reward maximization problem. In this way, the static, rule-based allocation strategy is replaced with a dynamic decision-making method for profit maximization. [10]. As RL methodology tries to obtain better policies during its training it needs not only to find better profit maximization

strategy, but also to comply with a given ruleset. This becomes a challenging task, as reward function needs to be differentiable and can contain contradicting constraints.

The training is described in Algorithm 1. During each training episode the current market state is observed and during each trading (simulation time) an action is selected based on current policy. The action is filtered based on predefined ruleset presented in Table 1.

Table 1

Rules and constraints set for the RL for OMX Baltics index stocks

Rule / constraint type	Declarative interpretation	Possible mathematical form
Budget constraint	The full capital must be allocated across the selected assets.	$\sum_{i=1}^n w_{i,t} = 1$
Long-only constraint	Short selling is not allowed; each asset weight must be non-negative.	$w_{i,t} \geq 0, \forall i$
Diversification constraint	No single asset may dominate the portfolio allocation.	$w_{i,t} \leq w_{\max}, \forall i$
Liquidity constraint	Assets with small trading volume should receive limited or zero weight.	$w_{i,t} \leq L_{i,t}$ or $w_{i,t} = 0$ if $Vol_{i,t} < Vol_{\min}$
Turnover constraint	Portfolio rebalancing should be limited to avoid excessive trading costs.	$\sum_{i=1}^n w_{i,t} - w_{i,t-1} \leq \tau_{\max}$
Risk constraint	The portfolio should remain below a predefined risk threshold.	$Var_t(w_t) \leq \zeta$ or $\sigma_{p,t} \leq \sigma_{\max}$
Country exposure constraint	Exposure to one Baltic country should not exceed a predefined limit.	$\sum_{i \in C_k} w_{i,t} \leq c_{\max}$
Sector exposure constraint	Exposure to one economic sector should not exceed a predefined limit.	$\sum_{i \in S_k} w_{i,t} \leq s_{\max}$

The scope of ruleset will be picked during experiment phase. After retrieving reward penalties the agent policy network is updated with the new reward information. After the training is done, the testing period will be chosen as a final check on the unseen data without network policy updates. Next, a final walkforward test will be performed that will test and train on multiple timeframes for more robust model validation.

In financial engineering an agent is a portfolio manager who seeks to maximize portfolio returns consistent with risk preferences. The action space of such an agent will be buy, sell, or hold as continuous based on appropriate weight allocation presented in Table 2.

Table 2

Proposed reinforcement learning formulation

Component	Description
State space s_t	The state represents the information about the environment to the agent before making a financial decision at time t . It can include historical prices, asset returns, volatility measures, trading volume, technical indicators, current portfolio weights, and possibly country or sector labels for the OMX Baltic assets.
Action space a_t	The action is a vector of portfolio weights assigned to the selected OMX Baltic assets: $a_t = (w_{1,t}, w_{2,t}, \dots, w_{n,t})$, where $w_{i,t}$ shows the proportion of capital allocated to asset i at time t . The agent can use continuous action space with preselected assets.
Reward function r_t	The reward is a measure of model performance under selected constraints. It can be defined as portfolio returns minus transaction costs and, if needed, risk-adjusted or constraint-violation terms. For example, $r_t = R_{p,t} - TC_t - \lambda V_t$ [2], where $R_{p,t}$ is portfolio return, TC_t is transaction cost, and V_t is a risk or rule-violation penalty.

Every time an agent performs an action that increases the portfolio reward, a positive reward signal is fired, and when there is a loss, a negative one is triggered. Using dynamic programming and the recursive Bellman equation for learning, an agent can converge on an optimal portfolio management policy (strategy) [9]. The DNN captures more complex aspects of the environment by using more hidden states. The drawback of such an approach is that it requires significant computational time and can

overfit to local policies, thus failing to explore valuable global strategies. RL can use several learning approaches: state-reward maximization, called quality learning or Q-learning; policy maximization, where the agent iterates over strategy sets to maximize rewards; actor-critic methods, which use a dual architecture of agents; and more.

The research plan can be outlined as follows:

1. Collect and preprocess historical market data of OMX Baltic asset universe. Define transaction costs.
2. Use training, validation, and testing periods.
3. Define the RL environment - state space, action space, reward function, and rule constraints.
4. Train the DDPG agent by sampling from replay buffer. Replay buffer is filled during exploration and training phase.
5. Update the actor and critic networks. Linear interpolation can be used.
6. Validate the model on a separate validation set for hyperparameter tuning.
7. Test the model on testing period with constant weights.
8. Evaluate the performance using different benchmarks. Compare with equal weighted portfolio, Mean-Variance portfolio and OMX Baltic index. Primary metrics - Sharpe ratio, Sortino ratio, total returns, number of rule violations, portfolio turnover.

Potential results would fill the gap of RL portfolio optimization case with rule-based constraints in the OMX Baltic market, providing research novelty and practical insights for portfolio managers.

4. Research Data

The initial asset universe consists of 49 shares listed on the Nasdaq Baltic regulated market [11], including 32 companies from the Baltic Main List and 17 companies from the Baltic Secondary List as presented in Table 3. A more detailed information on company tickers is presented in the Table 4. The research data will be based on daily price changes. First North securities are excluded because First North is an alternative market rather than a regulated market. For each security, the historical price series is collected from the first available observation in Nasdaq Baltic historical data up to the end of the sample period. Securities with insufficient history or liquidity may be excluded in a later liquidity-screened experimental universe.

Table 3

Summary of the Nasdaq Baltic share universe

Category	Number of companies
Tallinn market	20
Vilnius market	22
Riga market	7
Total	49

Table 4

Baltic regulated-market share universe considered for portfolio optimization

Vilnius market: 22 companies			Riga and Tallinn markets: 27 companies		
Ticker	Company	Market	Ticker	Company	Market
AKO1L	Akola Group	Vilnius	BAL1R	Amber Latvijas balzams	Riga
AMG1L	Amber Grid	Vilnius	DGR1R	DelfinGroup	Riga
APG1L	Apranga	Vilnius	ELEVR	Eleving Group	Riga
AUG1L	AUGA Group	Vilnius	IDX1R	INDEXO	Riga
GRG1L	Grigeo Group	Vilnius	RKB1R	Rigas kuģu būvētava	Riga
IGN1L	Ignitis grupė	Vilnius	SAF1R	SAF Tehnika	Riga
INC1L	INVL Technology	Vilnius	SCM1R	Siguldas CMAS	Riga
INL1L	INVL Baltic Farmland	Vilnius	ARC1T	Arco Vara	Tallinn
INR1L	INVL Baltic Real Estate	Vilnius	CPA1T	Coop Pank	Tallinn
IVL1L	Invalda INVL	Vilnius	EEG1T	Ekspress Grupp	Tallinn
KNE1L	KN Energies	Vilnius	EFT1T	EfTEN Real Estate Fund	Tallinn
KNR1L	Kauno energija	Vilnius	HAE1T	Harju Elekter Group	Tallinn
LGD1L	LITGRID	Vilnius	HPR1T	Hepsoor	Tallinn
NTU1L	Novaturas	Vilnius	INF1T	Infortar	Tallinn
PZV1L	Pieno žvaigždės	Vilnius	LHV1T	LHV Group	Tallinn
ROE1L	Artea Bankas	Vilnius	LVN1T	Liven	Tallinn
RSU1L	Rokiškio sūris	Vilnius	MRK1T	Merko Ehitus	Tallinn
TEL1L	Telia Lietuva	Vilnius	NCN1T	Nordecon	Tallinn
UTR1L	Utenos trikotažas	Vilnius	PKG1T	Pro Kapital Grupp	Tallinn
VBL1L	Vilniaus baldai	Vilnius	PRF1T	PRFoods	Tallinn
VLP1L	Vilkyškių pieninė	Vilnius	SFG1T	Silvano Fashion Group	Tallinn
ZMP1L	Žemaitijos pienas	Vilnius	SKN1T	Nordic Fibreboard	Tallinn
			TAL1T	Tallink Grupp	Tallinn
			TKM1T	TKM Grupp	Tallinn
			TPD1T	Trigon Property Development	Tallinn
			TSM1T	Tallinna Sadam	Tallinn
			TVE1T	Tallinna Vesi	Tallinn
Subtotal: 22 companies			Subtotal: 27 companies		

5. Experimental Setup

To obtain the results, the Vilniaus University high-performance computer cluster (HPC) is used with NVIDIA Tesla V100 for calculations. A FinRL framework [12] is used for portfolio optimization implementation in python programming language.

6. Research Novelty

The novelty of the work is based on the OMX Baltic market application with RL under rule-based constraints. The application of RL methodology with constraints in portfolio optimization is not fully researched in the scope of emerging markets and in the concrete example of the Baltic market. The emerging markets are different due to higher volatility and less data availability. Thus, the research has the potential to fulfill the gap.

7. Student Contribution

The student contribution is the idea proposal, experimental design, and implementation of the suggested research plan.

Declaration on Generative AI

AI tools were used solely for technical and editorial tasks. Specifically, Grammarly AI was employed for correcting grammar and style, ChatGPT for formatting LaTeX tables. Github Copilot was used for extracting information from *Nasdaq Baltic* website[11].

References

- [1] H. Markowitz, Portfolio Selection, *The Journal of Finance* 7 (1952) 77–91. URL: <https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1952.tb01525.x>. doi:10.1111/j.1540-6261.1952.tb01525.x.
- [2] B. Enkhsaikhan, O. Jo, Augmented Lagrangian Risk-constrained Reinforcement Learning for Portfolio Optimization (Student Abstract), *Proceedings of the AAAI Conference on Artificial Intelligence* 39 (2025) 29362–29364. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/35252>. doi:10.1609/aaai.v39i28.35252.
- [3] Y. Jiang, J. Olmo, M. Atwi, Deep reinforcement learning for portfolio selection, *Global Finance Journal* 62 (2024) 101016. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1044028324000887>. doi:10.1016/j.gfj.2024.101016.
- [4] D. Winkel, N. Strauß, M. Schubert, T. Seidl, Simplex Decomposition for Portfolio Allocation Constraints in Reinforcement Learning, in: K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, R. Rădulescu (Eds.), *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023. URL: <https://ebooks.iospress.nl/doi/10.3233/FAIA230573>. doi:10.3233/FAIA230573.
- [5] M. Gollart, Y. Okhrin, A reinforcement learning approach to dynamic portfolio optimization, *Annals of Operations Research* (2025). URL: <https://link.springer.com/10.1007/s10479-025-06649-x>. doi:10.1007/s10479-025-06649-x.
- [6] H. Beyhan, E. Ergin, B. S. Eren, Dynamic Portfolio Optimization with Deep Reinforcement Learning: Evidence from Borsa Istanbul, *Ekonomi Politika ve Finans Arastirmalari Dergisi* 11 (2026) 106–119. URL: <https://dergipark.org.tr/en/doi/10.30784/epfad.1811319>. doi:10.30784/epfad.1811319.
- [7] Y. Li, N. Du, X. Song, X. Yang, T. Cui, N. Xue, A. Farjudian, J. Ren, W. P. Cheah, Cardinality and Bounding Constrained Portfolio Optimization Using Safe Reinforcement Learning, in: *2024 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Yokohama, Japan, 2024, pp. 1–8. URL: <https://ieeexplore.ieee.org/document/10651491/>. doi:10.1109/IJCNN60899.2024.10651491.
- [8] Z. Li, V. Tam, K. L. Yeung, Investor Preference-Aware Portfolio Optimization with Deep Reinforcement Learning, in: *2025 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, Las Vegas, NV, USA, 2025, pp. 1–4. URL: <https://ieeexplore.ieee.org/document/10930205/>. doi:10.1109/ICCE63647.2025.10930205.
- [9] R. S. Sutton, A. G. Barto, *Reinforcement learning: an introduction*, Adaptive computation and machine learning series, second edition ed., The MIT Press, Cambridge, Massachusetts, 2018.
- [10] S. K. Sahu, A. Mokhade, N. D. Bokde, An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges, *Applied Sciences* 13 (2023). URL: <https://www.mdpi.com/2076-3417/13/3/1956>. doi:10.3390/app13031956.
- [11] Nasdaq, Baltic Nasdaq, 2026. URL: <https://nasdaqbaltic.com/>.
- [12] AI4Finance Foundation, FinRL Framework, 2026. URL: <http://github.com/AI4Finance-Foundation/FinRL>.