Exploring Modern Evolution Strategies in Portfolio Optimization

Ramin Hasani* RHASANI@CSAIL.MIT.EDU
Etan A. Ehsanfar* EHSANFAR@ICLOUD.COM,ETAN_EHSANFAR@VANGUARD.COM
Greg Banis
Rusty Bealer
Amir S. Ahmadi
Vanguard, United States

Abstract

Black-box optimization (BBO) techniques are often the core engine used in combinatorial optimization problems which include multi-asset class portfolio construction. The computational complexity of such evolutionary algorithms, however, is excessively high to the point that finding optimal portfolios in large search spaces becomes intractable and learning dynamics are usually heuristic. To alleviate these challenges, in this paper, we set out to leverage advances in meta-learning-based evolution strategy (ES), Adaptive ES-Active Subspaces, and fast-moving natural ES to improve high-dimensional portfolio construction. Using such modern ES algorithms in a series of risk-aware passive and active asset allocation problems, we obtain orders of magnitude efficiency in finding optimal portfolios compared to vanilla BBO methods. Moreover, as we increase the number of asset classes, our modern suite of BBOs finds better local optima resulting in better financial advice quality.

1. Introduction

Traditional methods of portfolio construction, including mean-variance optimization, operate based on two aspects: portfolio risk and portfolio return. However, actual investor decisions indicate that the selection of portfolios relies on a variety of risk and return dimensions, including systematic risk, volatility, active alpha, tracking error, and implicit risk factor exposures [1]. As a portfolio optimization problem, asset allocation could be solved by quadratic programs (QPs) through a set of linear constraints. However, the optimization problem becomes non-convex and computationally difficult when nonlinear constraints like transaction costs or minimum portfolio lots are incorporated [24]. Similarly, the inclusion of higher distribution moments in forecasts can result in added complexity [18]. In such circumstances, black-box optimization (BBO) algorithms that are gradient-free are great alternatives. In the space of portfolio optimization, BBO methods such as genetic algorithm (GA) are the go-to search strategies to find optimal portfolios [15, 23, 34]. Nonetheless, GA comes with a fundamental shortcoming that a simple GA algorithm on the OneMax problem* has *exponential runtime* with overwhelming probability for a population size $n \leq d^{1/8-\epsilon}$ for a problem size of d and a small $\epsilon > 0$ [30]. This implies that the application of GA to large-scale portfolio construction problems with large population sizes is computationally intractable.

*. These authors contributed equally to this work
*. The OneMax problem defines the task to find the binary string (of a given length) that maximizes the sum of its digits.

Recently, BBO alternatives such as parallelizable evolution strategies (ES) have shown promise in solving high-dimensional optimization problems comparable to gradient-based approaches [22, 36, 37]. These methods typically use first principles [31, 38] and require empirical tuning to find local optima. The evolutionary algorithms community has proposed numerous solutions to reducing the runtime complexity of ES [29], providing guidance to samples' update direction [7, 38], and informing the learning dynamics of the ES by meta-learning the search heuristics [21].

In the present work, we aim to extend these advances in BBO to portfolio optimization as we adopt multiple algorithms based on black-box evolutionary strategies to solve two general forms of utility maximization problems to find optimal portfolio asset allocation. In addition, we aim to utilitze the Jax library [4] to accelerate evolutionary computations including the utility function. Our experimental results mark a 10-1000 fold speed-up in searching for the optimal combination of assets. More importantly, we find better local optima in larger problems and spaces in comparison with the genetic algorithm. Our findings enable more efficient integration of BBO methods in portfolio management engines leading to better financial advice.

2. Related Works

In this section, we discuss research highly relevant to our work's scope.

Optimization in Portfolio Construction There are numerous optimization methods that can be used for portfolio construction. For instance, Black-box optimization (BBO) Methods such as GA, Particle Swarm Optimization (PSO), and Evolutionary Estrategies (ES) have been largely adopted in this space [1, 6, 12, 25–27, 33]. While BBO methods are preferable in non-convex settings and use cases where it is critical to solving for global minima, they are often compute-intensive and not scalable. More efficient algorithms such as dynamic programming (DP) and reinforcement learning (RL) could be used as an alternative to BBOs, as long as the problem is convex [3, 8, 9]. In asset management, many model-based and model-free RL and hybrid methods of those have been proposed [5, 11, 13, 17, 35, 39, 40].

Evolution Strategies (ES). ES [2, 32] is a popular BBO method that do not require access to gradients for optimization. This makes it applicable to nonconvex optimization problems, where solution candidates get selected, evaluated, and updated regardless of the fitness properties. ES typically refines a given distribution parameters via an iterative approach [14]. ES has been significantly improved over the last two decades to have their convergence, robustness to hyperparameters, complexity, and efficiency addressed [28, 38]. In particular, parallelization schemes allowed them to match the speed performance of gradient-based methods [36, 37].

In this work, we aim to show how modern ES algorithms could relax the assumptions of dynamic programming while improving the performance of BBOs. Accordingly, we adapt three recently developed algorithms namely simulation annealing, discovered ES (DES) [21], adaptive sample-efficient blackbox optimization via ES-active subspaces (ASEBO) [7], and cost-reduced fast-moving natural ES (CR-FM-NES) [29], to solve utility maximization problems for finding optimal distribution of assets for a given portfolio.

3. Background and Methodology

In this section, we describe the algorithms we explore in the context of utility maximization for asset allocation problems.

Given a function $f(x) : \mathcal{R}^D \rightarrow \mathcal{R}$, in a BBO the objective function is:

$$\min_{\mathbf{x}} f(x), x_d : \forall d \in 1, 2, ..., D$$

The Evolutionary Strategies (ES) assumes a search distribution to maximize the probability of high fitness for candidates in each generation. For instance, given a Gaussian distribution consists of a mean vector **m** $\in \mathcal{R}^D$ and a covariance matrix $\Sigma \in \mathcal{R}^{D\times D}$ where D denotes the number of search space dimensions. Additionally we harness accelerated computing by modifying *ask-evaluate-tell* API developed in EVOSAX [4, 20], a library that efficiently implemented evolutionary strategies in JAX. This interface iterates through through 1) **ask**: samples the candidate solutions, given a search distribution for D-dimensional space: $x_j \in \mathcal{R}^D \sim \mathcal{N}(m, \Sigma), \forall j = 1, 2, ..., N(strategy.ask(...))$, 2) **evaluate**: calculates the black-box fitness function for the selected candidates: $f(x_j)$, and 3) **tell**: updates the search distribution: $(m, \Sigma, x_j, f_j) \rightarrow m', \Sigma'$.

3.1. Genetic Algorithm

A genetic algorithm (GA) evolves $\mathcal{P}$ out of N individuals at every generation given the fitness score f_i with parameters (i.e. genotype) θ_i. The top T individuals become the parent of the next generation. In the Gaussian GA, a parent is selected uniformly at random with replacement, and Gaussian noise is added to the parameter vector: $\theta' = \theta + \sigma\epsilon,\ \epsilon \in \mathcal{N}(0, I)$ A common problem in this algorithm is convergence to sub-optimal results. In response, self-adaptive mutation rate (SAMR) and Group Elite Selection of Mutation Rates (GESMR) have been proposed [10, 19].

3.2. Simulated Annealing

The simulation annealing (SA) algorithm is based on the theorem developed in statistical physics and the Boltzmann distribution of system state given the system temperature [16].

3.3. Discovered Evolution Strategies Revisit

As a meta-learned black-box optimization approach, the DES [21] finds the best performing evolution strategy (ES) for functions without access to gradients which is invariant to the solutions (i.e.population) order using meta-learning. A self-attention architecture is suggested to address the variant solution order and the following rules update the mean and variance of the solution in an evolution strategy.

The population weights, ω_t, are updated using self-attention architecture:

$$\omega_t = softmax(softmax(\frac{QK^T}{\sqrt{D_K}})V)$$

where queries (Q), keys (K), and values (V) use learned linear transforms. This architecture is highly generalizable to unseen optimization tasks, supervised learning problems, and continuous control tasks.

3.4. Adaptive ES-Active Subspaces for Blackbox Optimization (ASEBO)

The main idea of ASEBO, similar to ES-type BBO models is that we do not require an accurate estimation of the gradient. ASEBO continuously learns the bias of the underlying low-dimensional model for approximating the gradients via compressed sensing and contextual bandit methods. For this purpose, ASEBO dynamically learns the dimension of the gradient space at different stages of optimization without external supervision. Accordingly, it is more sample efficient than other black-box optimization algorithms [7].

3.5. Cost Reduced FM Natural Evolution Strategies (CR-FM-NES

CR-FM-NES [29] is a variant of natural evolutionary strategies (NES) [38] for high-dimensional black-box optimization. This approach utilizes a representation of the covariance matrix instead of the full covariance matrix. This reduces the time and space complexity of the Fast Moving NES from cubic $\mathcal{O}(d^3)$ and quadratic $\mathcal{O}(d^2)$ to linear $\mathcal{O}(d)$ complexity.

The EvoSax implementation of the CR-FM-NES is called preconditioning finite difference (PFD) which extends the FD approach to ES by adding uncertainty in the estimation of the fitness gradient.

3.6. Search Distribution in portfolio optimization

The distribution from which we sample search parameters across all aforementioned BBO algorithms is typically the *normal* distribution. To prepare these search algorithms for multi-class portfolio optimization problems, we change the distribution to log-normal and divide the outcomes by the sum of all variables. This is to make sure the generated candidate portfolio weights are between 0 and 1 and sum to 1.

4. Portfolio Construction Experiment and Results

The asset allocation model aims to determine the optimal allocation of asset classes in a portfolio. The allocation model takes input from an investor such as her attitude toward passive and alpha risk etc., and also from a forward-looking Monte-Carlo-generated simulation of market returns for asset classes and time horizons.

In this section, we discuss results for optimizing the portfolios with various asset classes (ACs), trading strategies, population size, and algorithm generations. For each algorithm, the population size and the generations, and other optimization parameters, depending on the algorithm, are determined through hyper-tuning. The passive risk is assumed either 7 or 8 depending on the passive or active trading strategies.

4.1. Experiment One: Minimum Constraints

For passive portfolio construction, we define the following objective utility based on the prospective cumulative return of the portfolio in 10 years and the risk aversion rate (PRA):

$$Adjusted\ Return = \frac{1}{1-RA}[Cumulative\ Return]^{(1-RA)} \tag{1}$$

where for passive return, we assume $RA = 8$. The *cumulative return* in the above equation refers to the total return of the portfolio at the end of 10 years investment horizon.

We implement and compare the aforementioned algorithms with passive utility functions for 8 and 16 passive ACs as follows.

Table 1: Portfolio Optimization with 8 Asset Classes

Algorithm	Pop. Size	Generations	Search Cap.	Utility	Proc. Time (sec)
Genetic Algorithm	1000	100	100k	$-8.0427e{-4}$	204
Simulated Annealing	200	100	20k	$-8.0429e{-4}$	3.92
DES	835	178	148k	$\mathbf{-8.0426}e\mathbf{-4}$	3.6
ASEBO	100	450	45k	$-8.3151e{-4}$	7.68
CR-FM-NES	80	50	4k	$-8.043e{-4}$	6.9

The results for 8 ACs in Table 1 indicate that multiple alternatives of the genetic algorithm can solve the portfolio optimization to the same precision in a significantly faster time. Particularly, the DES obtains a better utility corresponding to a 98% improvement in processing time over the GA.

According to Table 2, the GA consumes 4 times as much time for solving the lowest utility for 16 ACs, which also provides worse utility compared to 8 ACs due to the possibly higher combinatorial complexity, nonetheless, DES still produces top results with 97% improvement in processing time. It is worthwhile to note that the DES and CR-FM-NES both can produce better results than 8 ACs in much faster time and the DES can produce utility results slightly worse than CR-FM-NES's under 10 seconds.

Table 2: Portfolio Optimization with 16 Asset Classes

Algorithm	Pop. Size	Generations	Search Cap.	Utility	Proc. Time (sec)
Genetic Algorithm	5000	100	500k	$-8.5114e{-4}$	840.34
Simulated Annealing	5000	100	500k	$-8.0093e{-4}$	4.93
DES	2110	905	1910k	$\mathbf{-7.8762}e{-4}$	23.27
ASEBO	500	50	25k	$-8.7351e{-4}$	4.33
CR-FM-NES	620	650	403k	$-7.9121e{-4}$	16.76

4.2. Experiment Two: Seeking Alpha

In this section, we add a minimal set of assumptions to seek higher alpha. The first assumptions is selecting few assets for active trading which are simply assumed as separate asset classes. Second, we add an expected return and the variance empirically extracted from tracking asset classes. Finally the active risk aversion rate is assumed 10 for active asset classes which directly impacts the return:

$$Adjusted\ Return = Adjusted\ Passive\ Return[RA = 8] + Adjusted\ Active\ Return[RA = 10]$$

Tables 3 and 4 show results with active constraints. For 8 passive and 4 active ACs, the ASEBO has surpassed the other three algorithms in producing the highest utility while DES dominates the table in terms of utility score. With the addition of another 8 assets for passive trading in Table 4, the overall utility has improved across the board nonetheless time complexity of the GA has deteriorated except while that of ASEBO and CR-FM-NES has improved with additional ACs. According to the portfolio weights, the complexity of the optimized portfolio weights across all algorithms is reduced with the addition of active ACs. Notice that adding active components to the problem elevates the scale of the utility scores as illustrated in Tables 3 and 4 compared to deviate passive-only portfolios presented in Tables 1 and 2. In addition, the simulation annealing (SA) algorithm is removed from these tables because of its intractable complexity for active trading.

Table 3: Portfolio Optimization with 10 Asset Classes

Algorithm	Pop. Size	Generations	Search Cap.	Utility	Proc. Time (sec)
Genetic Algorithm	1000	100	100k	-0.123435	269
DES	530	82	43k	-0.123466	5.86
ASEBO	140	740	103k	$\mathbf{-0.1234287}$	24.57
CR-FM-NES	80	830	66k	-0.1240545	29.71

Table 4: Portfolio Optimization with 20 Asset Classes

Algorithm	Pop. Size	Generations	Search Cap.	Utility	Proc. Time (sec)
Genetic Algorithm	5000	100	500k	-0.119986	1255
DES	530	82	43k	$\mathbf{-0.119005}$	6.94
ASEBO	140	740	103k	-0.119569	4.61
CR-FM-NES	80	830	66k	-0.119219	8.75

4.3. Conclusion

This paper introduced a Jax-optimized framework for a typical portfolio optimization problem with forward-looking asset return scenarios. We used GA as the baseline and compared the results of other algorithms such as SA, DES, ASEBO, and CR-FM-NES for different passive and active asset classes. The notable results show that the DES can improve the utility results over the baseline in all circumstances and simultaneously perform the optimization task in under 10 seconds which is over 98% percent improvement over the GA. Results indicate a strong case for portfolio construction using modern GPU-enabled evolutionary black-box optimization.

5. Notes

All investing is subject to risk, including the possible loss of the money you invest. Be aware that fluctuations in the financial markets and other factors may cause declines in the value of your account. There is no guarantee that any particular asset allocation or mix of funds will meet your investment objectives or provide you with a given level of income. This material is provided for informational purposes only and is not intended to be investment advice or a recommendation to take any particular investment action.

References

[1] Roger Aliaga-Díaz, Giulio Renzi-Ricci, A Daga, and Harshdeep Ahluwalia. Vanguard asset allocation model: An investment solution for active-passive-factor portfolios. *rn*, 10:9, 2019.

[2] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1:3–52, 2002.

[3] Stephen Boyd, Enzo Busseti, Steve Diamond, Ronald N Kahn, Kwangmoo Koh, Peter Nystrup, Jan Speth, et al. Multi-period trading via convex optimization. *Foundations and Trends® in Optimization*, 3(1):1–76, 2017.

[4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[5] Ayman Chaouki, Stephen Hardiman, Christian Schmidt, Emmanuel Sérié, and Joachim De Lataillade. Deep deterministic portfolio optimization. *The Journal of Finance and Data Science*, 6:16–30, 2020.

[6] Chen Chen and Yu-sha Zhou. Robust multiobjective portfolio with higher moments. *Expert Systems with Applications*, 100:165–181, 2018.

[7] Krzysztof M Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, and Vikas Sindhwani. From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Xiangyu Cui, Jianjun Gao, and Yun Shi. Multi-period mean–variance portfolio optimization with management fees. *Operational Research*, 21:1333–1354, 2021.

[9] Hanjun Dai, Yuan Xue, Zia Syed, Dale Schuurmans, and Bo Dai. Neural stochastic dual dynamic programming. *arXiv preprint arXiv:2112.00874*, 2021.

[10] Duc-Cuong Dang and Per Kristian Lehre. Self-adaptation of mutation rates in non-elitist populations. In *Parallel Problem Solving from Nature–PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings 14*, pages 803–813. Springer, 2016.

[11] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.

[12] Fernando García García, Jairo González-Bueno, Francisco Guijarro, Javier Oliver-Muncharaz, and Rima Tamosiuniene. Multiobjective approach to portfolio optimization in the light of the credibility theory. *Technological and Economic Development of Economy (Online)*, 26(6): 1165–1186, 2020.

[13] Maryam Ghandehari, Adel Azar, Ahmad Reza Yazdanian, and Gholamhossein Golarzi. A hybrid model of stochastic dynamic programming and genetic algorithm for multistage portfolio optimization with gluevar risk measurement. *Industrial Management Journal*, 11(3):517–542, 2019.

[14] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[15] Jh Holland. Adaptation in natural and artificial systems. *Ann Arbor*, 1975.

[16] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[17] Weiwei Kong, Christopher Liaw, Aranyak Mehta, and D Sivakumar. A new dog learns old tricks: Rl finds classic optimization algorithms. In *International conference on learning representations*, 2019.

[18] Saranya Kshatriya and P Krishna Prasanna. Genetic algorithm-based portfolio optimization with higher moments in global stock markets. *Journal of Risk*, 20(4), 2018.

[19] Akarsh Kumar, Bo Liu, Risto Miikkulainen, and Peter Stone. Effective mutation rate adaptation through group elite selection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 721–729, 2022.

[20] Robert Tjarko Lange. evosax: Jax-based evolution strategies. *arXiv preprint arXiv:2212.04180*, 2022.

[21] Robert Tjarko Lange, Tom Schaul, Yutian Chen, Tom Zahavy, Valenti Dallibard, Chris Lu, Satinder Singh, and Sebastian Flennerhag. Discovering evolution strategies via meta-black-box optimization. *arXiv preprint arXiv:2211.11260*, 2022.

[22] Kuang-Huei Lee, Ofir Nachum, Tingnan Zhang, Sergio Guadarrama, Jie Tan, and Wenhao Yu. Pi-ars: Accelerating evolution-learned visual-locomotion with predictive information representations. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1447–1454. IEEE, 2022.

[23] Chi-Ming Lin and Mitsuo Gen. An effective decision-based genetic algorithm approach to multiobjective portfolio optimization problem. *Applied mathematical sciences*, 1(5):201–210, 2007.

[24] Dan Lin, Xiaoming Li, and Minqiang Li. A genetic algorithm for solving portfolio optimization problems with transaction costs and minimum transaction lots. In *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part III 1*, pages 808–811. Springer, 2005.

[25] Luís Lobato Macedo, Pedro Godinho, and Maria João Alves. Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. *Expert Systems with Applications*, 79:33–43, 2017.

[26] Danilo Alcantara Milhomem and Maria José Pereira Dantas. Analysis of new approaches used in portfolio optimization: a systematic literature review. *Production*, 30, 2020.

[27] Sudhansu Kumar Mishra, Ganapati Panda, and Babita Majhi. Prediction based mean-variance model for constrained portfolio assets selection using multiobjective evolutionary algorithms. *Swarm and evolutionary computation*, 28:117–130, 2016.

[28] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

[29] Masahiro Nomura and Isao Ono. Fast moving natural evolution strategy for high-dimensional problems. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2022.

[30] Pietro S Oliveto and Carsten Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, 2014.

[31] Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *The Journal of Machine Learning Research*, 18(1):564–628, 2017.

[32] Ingo Rechenberg. Evolutionsstrategie. *Optimierung technischer Systeme nach Prinzipien derbiologischen Evolution*, 1973.

[33] Alejandro Reveiz-Herault. An active asset management investment process for drawdown-averse investors. *Intelligent Systems in Accounting, Finance and Management*, 23(1-2):85–96, 2016.

[34] Omar Rifki and Hirotaka Ono. A survey of computational approaches to portfolio optimization by genetic algorithms. In *18th International Conference Computing in Economics and Finance*. Society for Computational Economics, 2012.

[35] Gordon Ritter. Machine learning for trading. *Available at SSRN 3015609*, 2017.

[36] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[37] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

[38] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[39] Pengqian Yu, Joon Sern Lee, Ilya Kulyatin, Zekun Shi, and Sakyasingha Dasgupta. Model-based deep reinforcement learning for dynamic portfolio optimization. *arXiv preprint arXiv:1901.08740*, 2019.

[40] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2):25–40, 2020.