



Hyperparameter Optimization for Portfolio Selection

Nystrup, Peter; Lindström, Erik; Madsen, Henrik

Published in:
Journal of Financial Data Science

Link to article, DOI:
[10.3905/jfds.2020.1.035](https://doi.org/10.3905/jfds.2020.1.035)

Publication date:
2020

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Nystrup, P., Lindström, E., & Madsen, H. (2020). Hyperparameter Optimization for Portfolio Selection. *Journal of Financial Data Science*, 2(3), 40-54. <https://doi.org/10.3905/jfds.2020.1.035>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Hyperparameter Optimization for Portfolio Selection

Peter Nystrup is a postdoctoral fellow at the centre for mathematical sciences, Lund University, Box 118, 221 00 Lund, Sweden, and in the department of applied mathematics and computer science, Technical University of Denmark, Asmussens Allé, Building 303B, 2800 Kgs. Lyngby, Denmark.
peter.nystrup@matstat.lu.se +45 20420498

Erik Lindström is a professor and head of the section for mathematical statistics at the centre for mathematical sciences, Lund University, Box 118, 221 00 Lund, Sweden.
erik.lindstrom@matstat.lu.se +46 462224578

Henrik Madsen is a professor and head of the section for dynamical systems in the department of applied mathematics and computer science, Technical University of Denmark, Asmussens Allé, Building 303B, 2800 Kgs. Lyngby, Denmark.
hmad@dtu.dk +45 45253408

HIGHLIGHTS

- The growing number of applications of machine-learning approaches to portfolio selection means that hyperparameter optimization becomes increasingly important. We propose a systematic approach to hyperparameter optimization by leveraging recent advances in automated machine learning and multi-objective optimization.
- We establish a connection between forecast uncertainty and holding- and trading-cost parameters in portfolio selection. We argue that they should be considered regularization parameters that can be adjusted in training to achieve optimal performance when tested subject to realized costs.
- We show that multi-objective optimization can find solutions with better return/risk tradeoffs than manual, grid, and random search over hyperparameters for portfolio selection. At the same time, the solutions are more stable across in-sample training and out-of-sample testing.

Abstract

Portfolio selection involves a tradeoff between maximizing expected return and minimizing risk. In practice, useful formulations also include various costs and constraints that regularize the problem and reduce the risk due to estimation errors, resulting in solutions that depend on a number of hyperparameters. As the number of hyperparameters grows, selecting their value becomes increasingly important and difficult. In this article we propose a systematic approach to hyperparameter optimization by leveraging recent advances in automated machine learning and multi-objective optimization. We optimize hyperparameters on a train set to yield the best result subject to market-determined realized costs. In applications to single- and multi-period portfolio selection, we show that sequential hyperparameter optimization finds solutions with better return/risk tradeoffs than manual, grid, and random search over hyperparameters using fewer function evaluations. At the same time, the solutions found are more stable from in-sample training to out-of-sample testing, suggesting they are less likely to be extremities that randomly happened to yield good performance in training.

Since Markowitz (1952) formulated the portfolio selection problem as a *tradeoff* between maximizing expected return and minimizing risk, this has been the default approach in finance (Kolm, Tütüncü, and Fabozzi, 2014). Building financial models from the ground up that model many or all aspects of the portfolio selection problem is challenging. Specifically, each optimal submodel (e.g., transaction costs, holding costs, risk, return forecasts, etc.) is noisy and has its own approximation errors. We cannot expect that combining several noisy models together will be optimal. An alternative is to choose hyperparameters for each submodel based on other criteria, such as that of minimizing uncertainty or maximizing some performance metric (Ban, Karoui, and Lim, 2018).

Multi-period investment problems taking into account the stochastic nature of financial markets are usually solved in practice by scenario approximations of stochastic programming models, which are computationally challenging (see, e.g., Dantzig and In-fanger, 1993, Mulvey and Shetty, 2004, Gülpınar and Rustem, 2007, Pınar, 2007). Recent literature has explored simulation-and-regression approaches to approximate the optimal policy through a large number of repetitions using *reinforcement learning* (Denault, Delage, and Simonato, 2017, Zhang et al., 2018, Kolm and Ritter, 2019). Both approaches require a good simulator.

Over the last decade, there has been an increasing interest in control-based approaches to solving multi-period versions of the problem. This includes the use of *model predictive control* (MPC) for option hedging (Meindl and Primbs, 2008, Bemporad, Bellucci, and Gabbriellini, 2014), trade planning (Boyd et al., 2017), and portfolio selection (Herzog, Dondi, and Geering, 2007, Nystrup et al., 2019). The idea is to control a portfolio based on forecasts of asset returns and relevant parameters. It is an intuitive approach with potential in practical applications because it is computationally fast. This makes it feasible to consider large numbers of assets and implement important constraints and costs.

The MPC formulations include hyperparameters such as risk, trading, holding, and shorting aversions that can be adjusted to achieve good performance in backtest and stresstest. Similarly, several studies have considered applications of regularization tech-

niques to portfolio selection in the Markowitz (1952) framework (see, e.g., Clempner and Poznyak, 2018, Aldridge, 2019, Nystrup et al., 2019, and references therein) to address estimation errors that can significantly impact portfolio weights and result in poor out-of-sample performance. Including or amplifying transaction and holding costs and constraining portfolio weights are ways to regularize the optimization problem and reduce the risk due to estimation errors. For example, it can be beneficial to include a holding cost for short positions or a long-only constraint to limit or prevent short positions, even if the actual portfolio is not subject to such cost nor constraint (Jagannathan and Ma, 2003). In addition to the parameters in the portfolio selection problem, the set of hyperparameters could include hyperparameters in the forecasting model, such as memory and regularization parameters. We will leave the latter for future research to explore.

When the portfolio selection problem is convex and of the dimensions we consider in this article, it can be solved quickly using a custom solver. These solvers are fast enough to run in real time. The practical advantage of the high speed is the ability to carry out a large number of backtests quickly. Despite computational advances, it generally is not feasible to use brute force to find good values of the hyperparameters. Therefore, it is of interest to explore hyperparameter optimization in relation to portfolio selection.

We consider an innovative application of *multi-objective optimization* (MOO) to hyperparameter optimization and portfolio selection. We leverage recent advances in automated machine learning and apply them to problems of practical relevance in finance and portfolio selection. Through a combination of simulation and practical examples, we promote the idea of considering transaction and holding costs and constraints as surrogates or regularization parameters that can be adjusted in training to achieve optimal performance when tested subject to realized costs. Surprisingly, despite the work of Jagannathan and Ma (2003) alluding to this nearly two decades ago, the idea is not yet widespread within the field of finance.

Our work shows the potential value of hyperparameter optimization for both single- and multi-period portfolio selection problems. In the single-period example, using MOO it is possible to find better return/risk tradeoffs in less than half of the time compared to

using grid search. Moreover, to illustrate that transaction costs can reduce the risk due to estimation errors, we show empirically that when the signal-to-noise ratio of return forecasts deteriorates, the optimal level of transaction costs imposed in portfolio selection increases to prevent excessive noise trading. In the multi-period example, we compare the result of MOO to random search. Using the same number of function evaluations, MOO finds better return/risk tradeoffs on training data. To examine whether hyperparameter optimization increases the risk of overfitting, we evaluate the results on test data and find that the optimized efficient frontier is more stable than the random-search frontier.

We begin with a brief introduction to hyperparameter and multi-objective optimization. Then we consider a single-period portfolio selection example from Boyd et al. (2017, Section 7.3). Subsequently, we consider an application of MPC to multi-period portfolio selection based on Nystrup et al. (2019). Finally, we conclude and provide suggestions for future work.

HYPERPARAMETER OPTIMIZATION

Every machine learning system has hyperparameters. An important task in automated machine learning is to automatically set these hyperparameters to optimize performance. Recent interest in complex and computationally expensive machine learning models with many hyperparameters, such as deep neural networks, has resulted in a resurgence of research on hyperparameter optimization (Feurer and Hutter, 2019). Manual and grid search are the most widely used strategies, despite decades of research into global optimization (e.g., Nelder and Mead, 1965, Broyden, 1970, Powell, 1994) and the publishing of several specialized hyperparameter optimization algorithms (e.g., Bergstra et al., 2011, Hutter, Hoos, and Leyton-Brown, 2011, Snoek, Larochelle, and Adams, 2012, Loshchilov, Glasmachers, and Beyer, 2019).

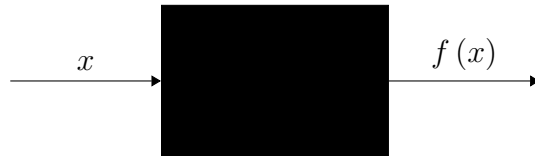
Manual search is the simplest approach and gives researchers some degree of insight into the optimization problem, but it often fails to find the optimum. Grid search and random search are extensions that are simple to implement and parallelize. *Grid search* is done by specifying a finite set of values for each hyperparameter and evaluating the

Cartesian product of these sets. It provides a systematic approach and typically finds a better solution than purely manual sequential optimization (in the same amount of time); but it scales poorly with the number of hyperparameters¹ and involves fitting the full model for values of hyperparameters that are very unpromising.

Random search has been proven to yield a faster exploration of the hyperparameter space than grid search, especially when some hyperparameters are much more important than others, which is often the case (Bergstra and Bengio, 2012). It is a useful method for initializing the search process, as it explores the entire configuration space and, thus, often finds settings with reasonable performance. Random search is a useful baseline because it makes no assumptions about the problem being optimized and, given enough resources, achieves performance arbitrarily close to the optimum. However, none of these methods make use of previous evaluations to make an informed decision of the next iterate. As such, convergence to a global optima can be very slow (Pedregosa, 2016).

In general, every *black-box* optimization method can be applied to hyperparameter optimization. In a black-box scenario, as illustrated in Exhibit 1, knowledge about an objective function $f : X \rightarrow \mathbb{R}$, to be optimized on some space X , is restricted to the handling of a device (e.g., simulation code) that delivers the value of $f(x)$ for any input $x \in X$.

Exhibit 1
Black-box scenario



In our case, the output values are return and risk in a backtest for given values of a set of hyperparameters. Running a backtest, say with daily trading over a five-year period, is often computationally demanding, meaning that each function evaluation is expensive. Gradient-based methods are not applicable, because it is a black-box scenario. Rather, we consider a state-of-the-art *evolution strategy*, which, similarly to Bayesian optimization, uses past function evaluations to update the distribution from which new candidate solutions are sampled (Voß, Hansen, and Igel, 2010). Additionally, the update

¹The required number of function evaluations grows exponentially with the dimension of the configuration space.

of the distribution parameters follows a natural gradient approximation of the expected objective-function value (Hansen and Ostermeier, 2001).

The purpose of automated hyperparameter optimization is to reduce the human effort necessary for applying machine learning, improve the performance of machine learning algorithms, and improve the reproducibility and fairness of scientific studies (Feurer and Hutter, 2019). Automated hyperparameter optimization is clearly more reproducible than manual search. In particular, different solutions can only be compared fairly if they all receive the same level of tuning for the problem at hand. The best strategy for how to avoid *overfitting* varies across hyperparameter optimization problems and remains an open problem. Automated approaches are a step in the right direction. As such, the objective of automated hyperparameter optimization extends to financial applications.

MULTI-OBJECTIVE OPTIMIZATION

In practical applications it is often necessary to trade off two or more objectives. For example, the Markowitz (1952) problem of maximizing return and minimizing risk is a MOO problem. A MOO problem can be formulated as

$$\begin{aligned} & \text{minimize} && (f_1(x), f_2(x), \dots, f_K(x)) \\ & \text{subject to} && x \in X, \end{aligned} \tag{1}$$

where $K \geq 2$ is the number of objectives and X is the feasible set of decision vectors. The feasible set is typically defined by some constraint functions. If some objective function is to be maximized, it is equivalent to minimize its negative.

Efficient frontier

In MOO there typically does not exist one feasible solution $x \in X$ that optimizes all objective functions simultaneously. Therefore, attention is paid to Pareto-optimal solutions; that is, solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives. The set of Pareto-optimal outcomes is often called the Pareto front or *efficient frontier*.

The goal of MOO is to find a diverse set of Pareto-optimal solutions, which provide insights into the tradeoffs between the objectives. When approaching a MOO problem by linearly aggregating all objectives into a scalar function, as it is customarily done in financial applications (e.g., Zenios, 2007, Chapter 3), each weighting of the objectives yields only a subset of Pareto-optimal solutions (usually only a single solution). Hence, various optimization trials with different aggregations become necessary to find a diverse set of Pareto-optimal solutions (Igel, Hansen, and Roth, 2007). For example, the standard approach in portfolio selection is to solve the Markowitz’ (1952) problem:

$$\begin{aligned} & \text{maximize} && \mu^T w - \gamma^{\text{risk}} w^T \Sigma w \\ & \text{subject to} && \mathbf{1}^T w = 1, \end{aligned} \tag{2}$$

where the variable w is the unknown portfolio weights, μ is the vector of expected returns, Σ the covariance matrix for the returns, and $\mathbf{1}$ is a column vector with all entries one, for several different values of the risk-aversion parameter γ^{risk} in order to trace the efficient frontier.

MO-CMA-ES

We use the multi-objective covariance-matrix adaptation evolution strategy (MO-CMA-ES) from Voß, Hansen, and Igel (2010) that is implemented in the Distributed Evolutionary Algorithms in Python (DEAP) framework by Fortin et al. (2012). It is a stochastic, derivative-free method for numerical optimization of difficult (non-convex, ill-conditioned, multi-modal, rugged, noisy) continuous optimization problems. We use the MO-CMA-ES because it is one of the best known and most competitive black-box optimization algorithms (Feurer and Hutter, 2019).

The MO-CMA-ES belongs to the class of evolutionary algorithms. In each generation (iteration) new individuals (candidate solutions, denoted as x) are generated by variation of the current parental individuals. Then, some individuals are selected to become the parents in the next generation based on their objective value $f(x)$. Over the generation sequence, individuals with better and better f -values are selected.

The way that past functions evaluations are used to update the parameters of the distribution from which new candidate solutions are sampled is different from Bayesian optimization. In an evolution strategy, new candidate solutions are sampled according to a multivariate Gaussian distribution. Dependencies between the variables are represented by a covariance matrix. Updates of the mean and covariance matrix are done by maximizing the likelihood of previously successful candidate solutions and search steps, respectively.

Adaptation of the covariance matrix amounts to learning a second-order model of the underlying objective function similar to the approximation of the inverse Hessian matrix in the Quasi-Newton method in classical optimization (e.g., Broyden, 1970, Wills and Schön, 2017). In contrast to most classical methods, fewer assumptions on the nature of the underlying objective function are made. Only the ranking between candidate solutions is exploited for learning the sample distribution and neither derivatives nor the function values themselves are required by the method (Hansen and Ostermeier, 2001).

SINGLE-PERIOD PORTFOLIO SELECTION

In this section, we consider the single-period portfolio selection example from Boyd et al. (2017, Section 7.3).

Problem formulation

Given a current portfolio of N risky assets and one risk-free asset with weights $w_t \in \mathbb{R}^{N+1}$, the single-period portfolio selection problem is to

$$\begin{aligned} \text{maximize} \quad & \hat{\mu}_{t+1|t}^T w_{t+1} - \gamma^{\text{trade}} \hat{\phi}_{t+1|t}^{\text{trade}} (w_{t+1} - w_t) - \gamma^{\text{hold}} \hat{\phi}_{t+1|t}^{\text{hold}} (w_{t+1}) \\ & - \gamma^{\text{risk}} w_{t+1}^T \hat{\Sigma}_{t+1|t} w_{t+1} \\ \text{subject to} \quad & \mathbf{1}^T w_{t+1} = 1, \end{aligned} \tag{3}$$

with variable $w_{t+1} \in \mathbb{R}^{N+1}$, where $\hat{\mu}_{t+1|t}$ and $\hat{\Sigma}_{t+1|t}$ are the forecasts made at time t of the returns and covariances at time $t+1$. Additionally, a maximum leverage constraint $\|(w_{t+1})_{1:N}\|_1 \leq L^{\max}$ is imposed with $L^{\max} = 3$, where $(w_{t+1})_{1:N} = ((w_{t+1})_1, \dots, (w_{t+1})_N)$

is the vector of risky-asset weights.

Transaction and holding costs In formulation (3), $\hat{\phi}^{\text{trade}}$ and $\hat{\phi}^{\text{hold}}$ are estimates of the actual, realized transaction- and holding-cost functions, although they could be arbitrary (convex) functions found to give good performance in backtest. The holding cost includes a charge of one basis point (bp) for all risky assets, and the transaction cost is volume and volatility dependent with an assumed bid–ask spread of five bps. We refer to Boyd et al. (2017, Section 7.3) for the specific details of the example.

Hyperparameters Formulation (3) includes three hyperparameters that need to be selected: 1) A risk-aversion parameter $\gamma^{\text{risk}} \geq 0$ that discourages holding portfolios with high variance. Alternatively, the expected return can be maximized subject to an upper limit on the portfolio variance, or the portfolio variance can be minimized subject to a lower limit on the expected return (Kolm, Tütüncü, and Fabozzi, 2014). 2) A trading-aversion factor $\gamma^{\text{trade}} \geq 0$ that scales the transaction-cost function. Increasing it will deter trading or reduce turnover; decreasing it will increase trading and turnover. It is even possible to think of $1/\gamma^{\text{trade}}$ as the number of periods over which transaction costs are amortized (Grinold, 2006). Keeping $\gamma^{\text{trade}} = 1$ would mean that we require trades to be profitable after a single period; otherwise we will not trade. 3) A holding-aversion parameter $\gamma^{\text{hold}} \geq 0$ that scales the holding-cost function to encourage, or discourage, holding positions that incur holding costs, such as short positions.

Simulation study

We follow Boyd et al. (2017) by considering an asset universe consisting of $N = 430$ stocks from the S&P 500 index with each backtest consisting of five years of daily trading from January 2012 through December 2016.

Forecasts

Return forecasts are constructed by adding zero-mean noise to realized returns of individual stocks from the S&P 500 index and then rescaling, to obtain return estimates

that would (approximately) minimize mean squared error.

For all t the return estimates for risky assets are

$$(\hat{\mu}_t)_{1:N} = \alpha ((r_t)_{1:N} + \epsilon_t), \quad (4)$$

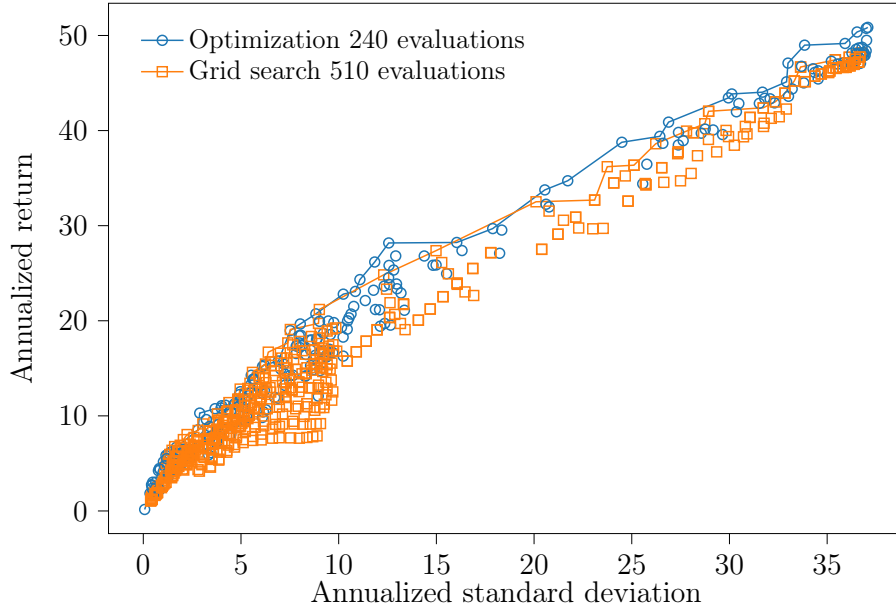
where $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2 I)$ are independent. We follow Boyd et al. (2017) and use noise variance $\sigma_\epsilon^2 = 0.02$, so the noise components have standard deviation around 14%, around a factor of 10 larger than the standard deviation of the realized returns. The scale factor α is chosen to minimize the mean squared error $E[(\hat{\mu}_t)_{1:N} - (r_t)_{1:N}]^2$, if we think of r_t as a random variable with variance σ_r , i.e., $\alpha = \sigma_r^2 / (\sigma_r^2 + \sigma_\epsilon^2)$. We use the typical value $\sigma_r^2 = 0.0005$, i.e., a realized return standard deviation of around 2%, so $\alpha = 0.024$. The typical return forecast is on the order of $\pm 0.3\%$. This corresponds to an information ratio $\sqrt{\alpha} \approx 0.15$, which is on the high end of what might be expected in practice (Grinold and Kahn, 2000).

Boyd et al. (2017) argue that with this level of noise and scaling, the return forecasts have an accuracy on the order of what might be expected from a proprietary forecast. For example, across all the assets and all days, the sign of predicted return agrees with the sign of the real return around 54% of the times. Later we will explore what happens when the noise variance σ_ϵ^2 is doubled to 0.04, but for now we will stick with the above values.

Total market volumes and daily volatilities (used in the transaction cost model) are estimated as moving averages of the realized values with a window of length 10. Finally, the covariance forecasts are constructed using a factor model estimated on a two-year rolling window of past realized returns. Once again we refer to Boyd et al. (2017, Section 7.3) for the specific details of the example.

Exhibit 2

Efficient frontiers based on grid search and multi-objective optimization



Results

After a couple of coarse grid searches to narrow in the region of interest, Boyd et al. (2017) performed a detailed hyperparameter search testing all combinations of

$$\gamma^{\text{risk}} = 0.1, 0.178, 0.316, 0.562, 1, 2, 3, 6, 10, 18, 32, 56, 100, 178, 316, 562, 1000,$$

$$\gamma^{\text{trade}} = 5.5, 6, 6.5, 7, 7.5, 8,$$

$$\gamma^{\text{hold}} = 0.1, 1, 10, 100, 1000,$$

a total of 510 backtest simulations. This takes about 12 hours on a standard laptop PC with a 1.9 GHz Intel i7 processor using its eight cores in parallel.

For comparison, we perform hyperparameter optimization by maximizing the return and minimizing the standard deviation using the MO-CMA-ES with initial step size $\sigma_0 = 0.1$. We map the above ranges for the hyperparameters to the unit interval $[0, 1]$ using power transformations. For example, $\gamma^{\text{risk}} = 10^{-1+4\theta_1}$, where θ_1 is the transformed risk-aversion parameter. Each generation consists of 20 individuals, and the first generation is sampled randomly from the three-dimensional unit cube.

Exhibit 2 compares the efficient frontier based on grid search to the result of MOO

using the MO-CMA-ES. With only 240 function evaluations compared to 510 using grid search, it is possible to find better return/risk tradeoffs using the MO-CMA-ES at a fraction of time. The computations involved in considering previous evaluations when sampling a new generation are negligible compared to the time it takes to run a five-year backtest. The objective values for an entire generation of hyperparameter values can be computed in parallel, similar to grid search. Hence, the overhead compared to grid search is minimal and it makes sense to compare the methods in terms of the number of backtest evaluations.

Holding aversion As noted by Boyd et al. (2017), we find that along the efficient frontier the optimal values of the holding-cost multiplier γ^{hold} (not shown) are much larger than one. This is an example of regularization in single-period portfolio selection. The large holding-aversion parameter tells the portfolio selection to avoid short positions, even though the actual holding cost (used by the simulator to update the portfolio each day) is only one bp, and the result is that the overall portfolio performance is better. This is consistent with the finding by Jagannathan and Ma (2003) that constraining portfolio weights to be nonnegative can reduce the risk in estimated optimal portfolios even when the constraints are wrong.

Different noise levels

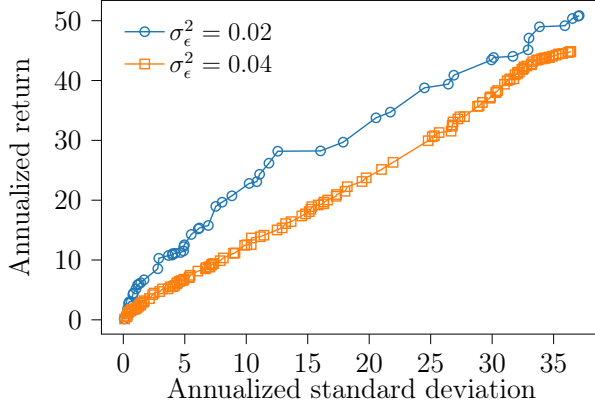
To further illustrate the benefits of hyperparameter optimization for single-period portfolio selection, we repeat the simulation study, only this time we double the variance of the noise added to the realized returns in order to construct the return forecasts from $\sigma_\epsilon^2 = 0.02$ to $\sigma_\epsilon^2 = 0.04$. This corresponds to an information ratio $\sqrt{\alpha} \approx 0.1$, i.e., by increasing the noise variance, the signal-to-noise ratio worsens.

Exhibit 3a shows the efficient frontiers based on MOO using 240 evaluations for the two different levels of noise in the return forecasts. The frontier for $\sigma_\epsilon^2 = 0.02$ is the same as in Exhibit 2. As expected, the weaker signal-to-noise ratio pushes the efficient frontier downward. With the added noise, the Sharpe ratios of the portfolios along the efficient frontier are closer to one than two.

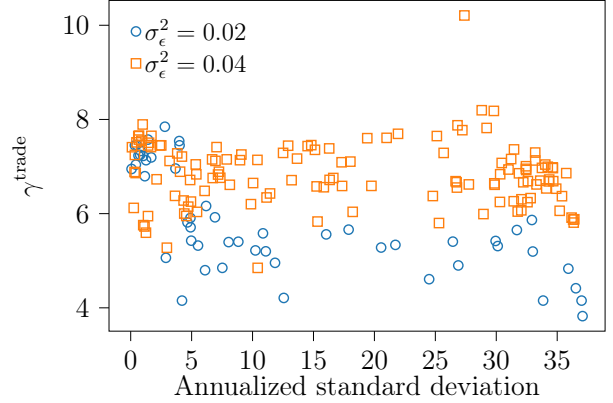
Exhibit 3

Efficient frontiers based on multi-objective optimization for different levels of noise in the return forecasts

(a) Efficient frontiers based on multi-objective optimization



(b) The optimal values of γ^{trade} for the points on the efficient frontiers



Note: The results are based on 240 function evaluations.

Exhibit 3b shows the optimal values of the trading-aversion parameter γ^{trade} for the points on the efficient frontiers. Although the optimal value varies with the risk level, it is clear that optimal values of the parameter are higher when the forecasts are more noisy. Roughly speaking, when the noise variance $\sigma_\epsilon^2 = 0.02$, the optimal values of γ^{trade} are around five, compared to around seven when $\sigma_\epsilon^2 = 0.04$. In other words, when the signal-to-noise ratio of the return forecasts deteriorates, the optimal level of the transaction-cost multiplier increases to prevent excessive trading, despite realized transaction costs remaining unchanged.

The optimization of the hyperparameters for two different levels of noise clearly shows the connection between the trading-aversion parameter and return forecast error. Similar to the holding-aversion parameter discussed above, the trading-aversion parameter regularizes the portfolio selection problem and reduces the risk due to estimation error. This would have been much harder to show without the use of MOO. Using grid search would have required significantly more function evaluations and at the same time we would have had to run extra checks to ensure that the grid was centered around the range of values of interest of the hyperparameters in order to see the differences. With a misaligned or too coarse grid we might not have been able to detect the differences.

It is both a strength and weakness that the hyperparameters in portfolio selection have

clear economic interpretations in terms of costs and constraints. It makes them intuitively easy to understand and justify, but at the same time it is easy to confuse them with their realized counterparts. We argue that the hyperparameters should be considered as regularization parameters that can be adjusted in training to achieve optimal performance when tested subject to realized costs. As illustrated in this section through different, realistic simulation examples, it is rarely optimal to keep the transaction- and holding-cost aversion parameters equal to one; that is, to use the best estimate of the realized costs in the portfolio selection. In the next section we will take this idea even further and consider trading- and holding-regularization functions that do not have to be a multiple of the estimated realized cost functions.

MULTI-PERIOD PORTFOLIO SELECTION

We continue by considering an application of MPC to multi-period portfolio selection based on Nystrup et al. (2019), where return and covariance forecasts are generated using an actual forecasting model. In a multi-period framework, we can naturally handle time-varying forecasts, market impact costs, and other phenomena that require planning multiple steps ahead (Boyd et al., 2017). To examine whether hyperparameter optimization increases the risk of overfitting, we compare results using both training and test data.

Model predictive control

MPC is based on the simple idea that, to determine the trades to make, all future (unknown) quantities are replaced by their forecasted values over a planning horizon H . For example, future returns are replaced by their expected values $\hat{\mu}_{\tau|t}$, $\tau = t+1, \dots, t+H$, where $\hat{\mu}_{\tau|t}$ is the forecast made at time t of the return at time τ . This leads to a deterministic optimization problem:

$$\begin{aligned} & \text{maximize} && \sum_{\tau=t+1}^{t+H} \left(\hat{\mu}_{\tau|t}^T w_{\tau} - \hat{\phi}_{\tau|t}^{\text{trade}}(w_{\tau} - w_{\tau-1}) - \hat{\phi}_{\tau|t}^{\text{hold}}(w_{\tau}) - \gamma^{\text{risk}} w_{\tau}^T \hat{\Sigma}_{\tau|t} w_{\tau} \right) \\ & \text{subject to} && \mathbf{1}^T w_{\tau} = 1, \quad \tau = t+1, \dots, t+H, \end{aligned} \quad (5)$$

with variables $w_{t+1}, \dots, w_{t+H} \in \mathbb{R}^{N+1}$. Note that w_t is not a variable but rather the known, current portfolio weights.

Solving the problem (5) yields an optimal sequence of weights $w_{t+1}^*, \dots, w_{t+H}^*$. The difference of this sequence is a plan for future trades over the planning horizon $H = 15$ under the unrealistic assumption that all future (unknown) quantities will be equal to their forecasted values. Only the first trade $w_{t+1}^* - w_t$ in the planned sequence of trades is executed. In the next step, the process is repeated, starting from the new portfolio w_{t+1} . The planning horizon can typically be much shorter than the investment horizon without affecting the solution (Boyd et al., 2017). MPC for stochastic systems is a suboptimal control strategy; however, it uses new information advantageously and is better than pure open-loop control.

Constraints

We consider long-only portfolios with maximum holdings of 40% per risky asset; i.e., $\mathbf{0} \leq (w_\tau)_{1:N} \leq (0.4) \mathbf{1}$, $\tau = t + 1, \dots, t + H$. Imposing constraints on the portfolio weights is a way to improve out-of-sample performance. It is equivalent to shrinking the covariance matrix (Jagannathan and Ma, 2003, Ledoit and Wolf, 2003, 2004, DeMiguel et al., 2009, Li, 2015). Different constraints correspond to different prior beliefs about the asset weights.

Additionally, the portfolios are subject to a maximum leverage constraint $\|(w_{t+1})_{1:N}\|_1 \leq L^{\max}$ with $L^{\max} = 2$. We assume that the only cost associated with leveraging a portfolio is the risk-free rate. We follow Nystrup et al. (2019) in specifying these holding and leverage constraints, although we note that the minimum and maximum holdings as well as the leverage limit could be considered hyperparameters to be selected along with the risk-aversion and cost parameters discussed next. This would render grid search infeasible for all practical purposes.

Transaction and holding penalties

As in the single-period setting, $\hat{\phi}^{\text{trade}}$ and $\hat{\phi}^{\text{hold}}$ can be estimates of actual transaction- and holding-cost functions or arbitrary (convex) regularization functions found to give good performance in backtest. This time we consider the latter approach. That is, we explore different trading- and holding-regularization functions while assuming a realized transaction cost of 10 bps per one way transaction and no realized holding cost when evaluating performance in backtests.

In order to regularize the portfolio selection problem and reduce the risk of trading too much, Nystrup et al. (2019) included a penalty for trading,

$$\phi_t^{\text{trade}}(w_t - w_{t-1}) = \kappa_1 \|w_t - w_{t-1}\|_1 + \kappa_2 \|w_t - w_{t-1}\|_2^2, \quad (6)$$

in the objective function, where κ_1 and κ_2 are penalty factors. This could reflect actual transaction costs or a conservatism toward trading, for example, due to the uncertainty related to the parameter estimates and forecasts.

The weighted elastic-net trading penalty (6) is a convex combination of ℓ_1 - and squared ℓ_2 -norm penalties. It reduces the number of trades like the ℓ_1 penalty and the size of trades like the squared ℓ_2 penalty. The ℓ_1 penalty is similar to the standard proportional transaction cost (which we assume when evaluating performance) and is a convex relaxation of constraining the number of trades. The squared ℓ_2 penalty is used to model price impact (Almgren and Chriss, 2001, Boyd et al., 2017); it shrinks together trades in correlated assets and splits trades over multiple days.²

Furthermore, Nystrup et al. (2019) included a holding penalty,

$$\phi_t^{\text{hold}}(w_t) = \rho_1 \|w_t\|_1 + \rho_2 \|w_t\|_2^2, \quad (7)$$

where ρ_1 and ρ_2 are penalty factors. Including holding costs is a means to control portfolio leverage, as discussed in the previous section. When considering long–short portfolios,

²Price impact is the price movement against the trader that tends to occur when a large order is executed.

this could include an additional holding penalty specific to short positions, which would be another hyperparameter.

The weighted elastic-net holding penalty (7) can be justified by reformulating the mean–variance criterion as a robust optimization problem (Ho, Sun, and Xin, 2015, Boyd et al., 2017). It reduces the number of holdings like the ℓ_1 penalty and the size of holdings like the squared ℓ_2 penalty. The ℓ_1 penalty is a convex relaxation of constraining the number of holdings. It can be regarded as a shrinkage estimator of the expected return (Stein, 1956, Fabozzi, Huang, and Zhou, 2010). The squared ℓ_2 penalty shrinks together holdings in correlated assets; it corresponds to adding a diagonal matrix to the forecasted covariance matrix in (5), similar to a Stein-type shrinkage estimator (Ledoit and Wolf, 2004). Even when a portfolio is not subject to any holding cost, it can still be beneficial to include holding penalties in the optimization problem.

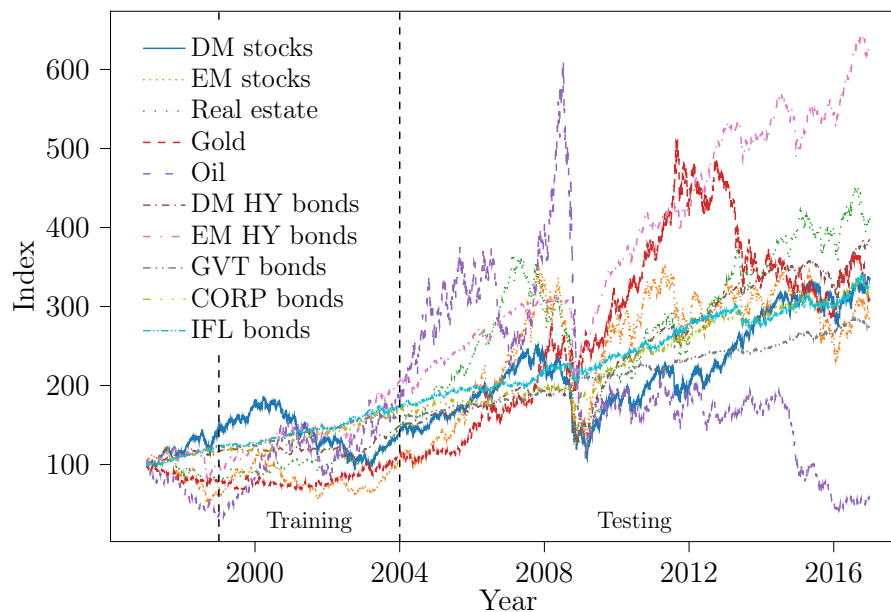
Considering the specific parameters in the trading- and holding-regularization functions as hyperparameters is a flexible alternative to introducing trading- and holding-aversion parameters that scale the realized cost functions. In total, the hyperparameter set consists of a risk-aversion parameter γ^{risk} ; trading-penalty factors κ_1 and κ_2 ; and holding-penalty factors ρ_1 and ρ_2 . Similar to Nystrup et al. (2019), we assume that trading- and holding-penalty parameters are the same for all risky assets and zero for the risk-free asset. Without this assumption, the hyperparameter set would be considerably larger and grid search would be completely infeasible.

Data

Exhibit 4 shows the daily index data considered by Nystrup et al. (2019). The 10 indices are DM and EM stocks, listed DM real estate, DM and EM high-yield bonds, gold, oil, corporate bonds, inflation-linked bonds, and government bonds. All indices measure the net total return in USD with a total of 5,185 daily closing prices per index covering the period from 1997 through 2016.

We use the first two years for model initialization, the next five years for training, and the last 13 years for testing. As always when considering financial data, the split in

Exhibit 4 Daily index data used for training and testing



train and test set is pivotal. Most importantly, the five years from 1999 to 2003 include both a bull and bear market. We use a different split than Nystrup et al. (2019), who used data prior to 1999 for training.

We use their forecasts of μ and Σ , which are based on a multivariate, Gaussian hidden Markov model with two states estimated using a sequential version of the Expectation–Maximization algorithm with a fading memory and without any forward looking. Every day when a new observation is recorded, the model estimates the probability of that observation belonging to the risk–on and risk–off states, respectively. Based on the estimated probabilities the model parameters are updated, and subsequently μ and Σ are forecasted 15 days into the future starting from the estimated probability distribution over the current market state. Several other studies have considered asset allocation based on regimes inferred from financial returns using regime-switching models (Guidolin and Timmermann, 2007, Bulla et al., 2011, Nystrup et al., 2015, 2017, Nystrup, Madsen, and Lindström, 2018).

Results

In this section, we focus on comparing the results of MOO and random search; that is, independent draws from a uniform density from the same configuration space as would be spanned by a regular grid. Bergstra and Bengio (2012) showed empirically and theoretically that randomly chosen trials are more efficient for hyperparameter optimization than trials on a grid. Thus, random search is a natural baseline against which to compare adaptive (sequential) hyperparameter optimization algorithms.

The portfolio selection problem (5) is solved using CVXPY (Diamond and Boyd, 2016) with the open-source solver ECOS (Domahidi, Chu, and Boyd, 2013). We perform hyperparameter optimization by maximizing the return and minimizing the maximum drawdown (MDD)³ over the five-year training period using the MO-CMA-ES with initial step size $\sigma_0 = 0.1$. Note that minimizing MDD is a different objective than in (5). As already discussed, it can be advantageous to not consider the same objective function as in the underlying model when optimizing hyperparameters.

As in the single-period setting, we map the regions of interest for the hyperparameters to the unit interval using power transformations: $\gamma^{\text{risk}} = 10^{-3+5\theta_1}$, $\kappa_1 = 10^{-7+6\theta_2}$, $\kappa_2 = 10^{-7+6\theta_3}$, $\rho_1 = 10^{-7+5\theta_4}$, and $\rho_2 = 10^{-7+5\theta_5}$. These intervals are selected based on initial experimentation and are wider than those considered by Nystrup et al. (2019). Each generation consists of 20 individuals, and the first generation is sampled randomly from the five-dimensional unit cube.

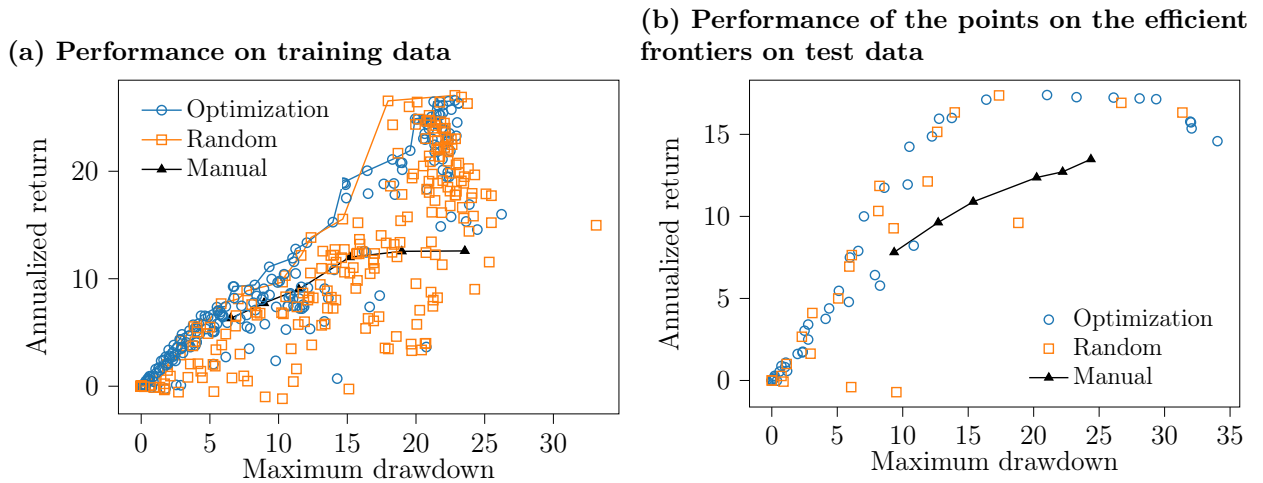
Training

Exhibit 5a compares the efficient frontiers based on random search and MOO using 240 evaluations on training data. The annualized return is in excess of the risk-free rate, which is the daily equivalent of the yield on a one-month U.S. treasury bill. The frontiers are compared to the frontier found by Nystrup et al. (2019), who did a coarse manual search in combination with linear aggregation for $\gamma^{\text{risk}} = 1, 3, 5, 10, 15, 25$. Both random search and MOO using the MO-CMA-ES find better return/risk tradeoffs than

³The maximum drawdown is the largest relative decline from a historical peak in the index value.

Exhibit 5

Efficient frontiers based on random search and multi-objective optimization compared to manual search



the points found by Nystrup et al. (2019). The MO-CMA-ES yields slightly better results than random search for MDDs smaller than 15%. It is evident that the search points are much more scattered for random search compared to the MO-CMA-ES.

The excess return that can be achieved for the same level of maximum drawdown compared to the manual search solutions amounts to several percentage points. Although it is interesting to see just how large the difference can be in a realistic application based on real forecasts, the value of hyperparameter optimization depends on the particular application, the number of hyperparameters, computational constraints, forecast strength, etc.

Testing

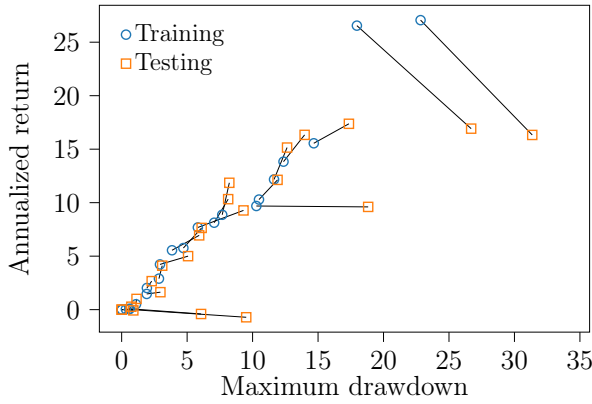
An important concern is to what extent the excess performance is a result of overfitting and, hence, will not persist when tested out of sample. After all, it seems reasonable to assume that optimizing the hyperparameters increases the risk of overfitting. This presumption was the reason why Nystrup et al. (2019) did not consider hyperparameter optimization in greater detail.

In Exhibit 5b, the points on the efficient frontiers in Exhibit 5a are evaluated on the test data. Once again, random search and MOO using the MO-CMA-ES yield

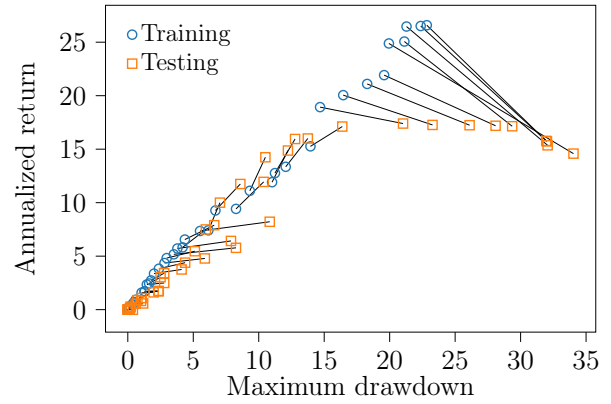
Exhibit 6

Connection between the points on the efficient frontier based on training data and their performance on test data

(a) Random search



(b) Multi-objective optimization



significantly better return/risk tradeoffs than manual search. This difference refutes the argument that a coarse manual or grid search decreases the risk of overfitting and increases out-of-sample robustness. The majority of the points lie close together for random search and the MO-CMA-ES, with the important exception of a handful of points that deviate considerably in a negative direction for random search.

Exhibit 6 illustrates the connection between the points on the efficient frontiers based on training data and their performance on the test data. Most of the points lie fairly close together. Yet, for random search, a few of the points that had very low MDDs in sample experience substantially higher MDDs out of sample. For the MO-CMA-ES, it is only the case for the points that had the highest MDDs on the training data that they realize substantially higher MDDs in testing. However, the mapping is consistent and suggests that risk taking above MDDs of 15% is not rewarded in the test period.

Although in sample the efficient frontier based on MOO was only slightly better than that based on random search, we notice important differences out of sample. In particular, the mapping from training to testing of the points on the efficient frontier is less stable when based on random search, as is evident from Exhibit 6.

In summary, the excess performance achieved through more careful hyperparameter optimization seems to persist from the in-sample training to the out-of-sample testing. Naturally, this finding is specific to the particular case, data period, train/test split, etc.

Nevertheless, it is an important finding that shows the potential for adding value both in and out of sample through a systematic approach to hyperparameter optimization.

CONCLUSION

The growing number of applications of machine-learning approaches to portfolio selection means that hyperparameter optimization becomes increasingly important. Through a number of examples, we illustrated the connection between forecast uncertainty and holding- and trading-regularization parameters in portfolio selection. In particular, we showed that when the signal-to-noise ratio of return forecasts deteriorates, the optimal level of the transaction-cost multiplier increases to prevent excessive trading, although realized transaction costs remain unchanged.

We showed that MOO can find solutions with better return/risk tradeoffs than manual, grid, and random search over hyperparameters. At the same time, solutions found using MOO were more stable across in-sample training and out-of-sample testing, indicating that improvements achieved through hyperparameter optimization were not a result of overfitting.

The value of hyperparameter optimization depends on the particular application, the number of hyperparameters, computational constraints, forecast strength, etc. However, using a sequential hyperparameter optimization algorithm, it is less likely that solutions found are extremities that randomly happened to yield good performance in training.

Future work involving hyperparameter optimization in relation to portfolio selection should consider MOO as an alternative to the more common methods of manual and grid search. Additionally, as hyperparameter optimization becomes increasingly important in quantitative finance and financial data science, future work should explore systematic approaches to combat overfitting.

ACKNOWLEDGMENTS

This work was supported by Vergstiftelsen and the Centre for IT-Intelligent Energy Systems (CITIES) project funded in part by Innovation Fund Denmark under Grant No.

1305-00027B. We are indebted to Ilya Loshchilov for proposing this article.

REFERENCES

- Aldridge, I. “Big Data in Portfolio Allocation: A New Approach to Successful Portfolio Optimization.” *Journal of Financial Data Science*, Vol. 1, No. 1 (2019), pp. 45–63.
- Almgren, R. and N. Chriss. “Optimal execution of portfolio transactions.” *Journal of Risk*, Vol. 3, No. 2 (2001), pp. 5–39.
- Ban, G.Y., N.E. Karoui, and A.E.B. Lim. “Machine Learning and Portfolio Optimization.” *Management Science*, Vol. 64, No. 3 (2018), pp. 1136–1154.
- Bemporad, A., L. Bellucci, and T. Gabbriellini. “Dynamic option hedging via stochastic model predictive control based on scenario simulation.” *Quantitative Finance*, Vol. 14, No. 10 (2014), pp. 1739–1751.
- Bergstra, J. and Y. Bengio. “Random Search for Hyper-Parameter Optimization.” *Journal of Machine Learning Research*, Vol. 13 (2012), pp. 281–305.
- Bergstra, J.S., R. Bardenet, Y. Bengio, and B. Kégl. “Algorithms for hyper-parameter optimization.” In *Advances in Neural Information Processing Systems* (2011), pp. 2546–2554.
- Boyd, S., E. Busseti, S. Diamond, R.N. Kahn, K. Koh, P. Nystrup, and J. Speth. “Multi-Period Trading via Convex Optimization.” *Foundations and Trends in Optimization*, Vol. 3, No. 1 (2017), pp. 1–76.
- Broyden, C.G. “The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations.” *IMA Journal of Applied Mathematics*, Vol. 6, No. 1 (1970), pp. 76–90.
- Bulla, J., S. Mergner, I. Bulla, A. Sesboüé, and C. Chesneau. “Markov-switching asset allocation: Do profitable strategies exist?” *Journal of Asset Management*, Vol. 12, No. 5 (2011), pp. 310–321.
- Clempner, J.B. and A.S. Poznyak. “Sparse mean–variance customer Markowitz portfolio optimization for Markov chains: a Tikhonov’s regularization penalty approach.” *Optimization and Engineering*, Vol. 19, No. 2 (2018), pp. 383–417.
- Dantzig, G.B. and G. Infanger. “Multi-stage stochastic linear programs for portfolio optimization.” *Annals of Operations Research*, Vol. 45, No. 1 (1993), pp. 59–76.
- DeMiguel, V., L. Garlappi, F. Nogales, and R. Uppal. “A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms.” *Management Science*, Vol. 55, No. 5 (2009), pp. 798–812.

- Denault, M., E. Delage, and J.G. Simonato. “Dynamic portfolio choice: a simulation-and-regression approach.” *Optimization and Engineering*, Vol. 18, No. 2 (2017), pp. 369–406.
- Diamond, S. and S. Boyd. “CVXPY: A Python-embedded modeling language for convex optimization.” *Journal of Machine Learning Research*, Vol. 17, No. 83 (2016), pp. 1–5.
- Domahidi, A., E. Chu, and S. Boyd. “ECOS: An SOCP Solver for Embedded Systems.” In *Proceedings of the 12th European Control Conference* (2013), pp. 3071–3076.
- Fabozzi, F.J., D. Huang, and G. Zhou. “Robust portfolios: contributions from operations research and finance.” *Annals of Operations Research*, Vol. 176, No. 1 (2010), pp. 191–220.
- Feurer, M. and F. Hutter. “Hyperparameter Optimization.” In F. Hutter, L. Kotthoff, and J. Vanschoren, editors, *Automated Machine Learning*, The Springer Series on Challenges in Machine Learning, chap. 1. Springer: Cham (2019), pp. 3–33.
- Fortin, F.A., F.M. De Rainville, M.A. Gardner, M. Parizeau, and C. Gagné. “DEAP: Evolutionary Algorithms Made Easy.” *Journal of Machine Learning Research*, Vol. 13 (2012), pp. 2171–2175.
- Grinold, R.C. “A Dynamic Model of Portfolio Management.” *Journal of Investment Management*, Vol. 4, No. 2 (2006), pp. 5–22.
- Grinold, R.C. and R.N. Kahn. *Active Portfolio Management: A Quantitative Approach for Providing Superior Returns and Controlling Risk*. McGraw–Hill: New York, 2nd ed. (2000).
- Guidolin, M. and A. Timmermann. “Asset allocation under multivariate regime switching.” *Journal of Economic Dynamics and Control*, Vol. 31, No. 11 (2007), pp. 3503–3544.
- Gülpınar, N. and B. Rustem. “Worst-case robust decisions for multi-period mean–variance portfolio optimization.” *European Journal of Operational Research*, Vol. 183, No. 3 (2007), pp. 981–1000.
- Hansen, N. and A. Ostermeier. “Completely derandomized self-adaptation in evolution strategies.” *Evolutionary Computation*, Vol. 9, No. 2 (2001), pp. 159–195.
- Herzog, F., G. Dondi, and H.P. Geering. “Stochastic model predictive control and portfolio optimization.” *International Journal of Theoretical and Applied Finance*, Vol. 10, No. 2 (2007), pp. 203–233.
- Ho, M., Z. Sun, and J. Xin. “Weighted elastic net penalized mean–variance portfolio design and computation.” *SIAM Journal on Financial Mathematics*, Vol. 6, No. 1 (2015), pp. 1220–1244.

- Hutter, F., H.H. Hoos, and K. Leyton-Brown. “Sequential Model-Based Optimization for General Algorithm Configuration.” In *Learning and Intelligent Optimization*, Vol. 6683 of *Lecture Notes in Computer Science*. Springer: Berlin (2011), pp. 507–523.
- Igel, C., N. Hansen, and S. Roth. “Covariance matrix adaptation for multi-objective optimization.” *Evolutionary Computation*, Vol. 15, No. 1 (2007), pp. 1–28.
- Jagannathan, R. and T. Ma. “Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps.” *Journal of Finance*, Vol. 58, No. 4 (2003), pp. 1651–1683.
- Kolm, P., R. Tütüncü, and F. Fabozzi. “60 Years of portfolio optimization: Practical challenges and current trends.” *European Journal of Operational Research*, Vol. 234, No. 2 (2014), pp. 356–371.
- Kolm, P.N. and G. Ritter. “Dynamic Replication and Hedging: A Reinforcement Learning Approach.” *Journal of Financial Data Science*, Vol. 1, No. 1 (2019), pp. 159–171.
- Ledoit, O. and M. Wolf. “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection.” *Journal of Empirical Finance*, Vol. 10, No. 5 (2003), pp. 603–621.
- . “A well-conditioned estimator for large-dimensional covariance matrices.” *Journal of Multivariate Analysis*, Vol. 88, No. 2 (2004), pp. 365–411.
- Li, J. “Sparse and Stable Portfolio Selection With Parameter Uncertainty.” *Journal of Business & Economic Statistics*, Vol. 33, No. 3 (2015), pp. 381–392.
- Loshchilov, I., T. Glasmachers, and H.G. Beyer. “Large Scale Black-Box Optimization by Limited-Memory Matrix Adaptation.” *IEEE Transactions on Evolutionary Computation*, Vol. 23, No. 2 (2019), pp. 353–358.
- Markowitz, H. “Portfolio Selection.” *Journal of Finance*, Vol. 7, No. 1 (1952), pp. 77–91.
- Meindl, P.J. and J.A. Primbs. “Dynamic hedging of single and multi-dimensional options with transaction costs: a generalized utility maximization approach.” *Quantitative Finance*, Vol. 8, No. 3 (2008), pp. 299–312.
- Mulvey, J.M. and B. Shetty. “Financial planning via multi-stage stochastic optimization.” *Computers & Operations Research*, Vol. 31, No. 1 (2004), pp. 1–20.
- Nelder, J.A. and R. Mead. “A Simplex Method for Function Minimization.” *Computer Journal*, Vol. 7, No. 4 (1965), pp. 308–313.
- Nystrup, P., S. Boyd, E. Lindström, and H. Madsen. “Multi-Period Portfolio Selection with Drawdown Control.” *Annals of Operations Research*, Vol. 282, No. 1–2 (2019), pp. 245–271.

- Nystrup, P., B.W. Hansen, H.O. Larsen, H. Madsen, and E. Lindström. “Dynamic Allocation or Diversification: A Regime-Based Approach to Multiple Assets.” *Journal of Portfolio Management*, Vol. 44, No. 2 (2017), pp. 62–73.
- Nystrup, P., B.W. Hansen, H. Madsen, and E. Lindström. “Regime-Based Versus Static Asset Allocation: Letting the Data Speak.” *Journal of Portfolio Management*, Vol. 42, No. 1 (2015), pp. 103–109.
- Nystrup, P., H. Madsen, and E. Lindström. “Dynamic portfolio optimization across hidden market regimes.” *Quantitative Finance*, Vol. 18, No. 1 (2018), pp. 83–95.
- Pedregosa, F. “Hyperparameter optimization with approximate gradient.” In *33rd International Conference on Machine Learning* (2016), pp. 737–746.
- Pinar, M.Ç. “Robust scenario optimization based on downside-risk measure for multi-period portfolio selection.” *OR Spectrum*, Vol. 29, No. 2 (2007), pp. 295–309.
- Powell, M.J.D. “A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation.” In *Advances in Optimization and Numerical Analysis*. Springer: Dordrecht (1994), pp. 51–67.
- Snoek, J., H. Larochelle, and R.P. Adams. “Practical Bayesian optimization of machine learning algorithms.” In *Advances in Neural Information Processing Systems* (2012), pp. 2951–2959.
- Stein, C. “Inadmissibility of the Usual Estimator for the Mean of a Multivariate Normal Distribution.” In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. University of California Press: Berkeley (1956), pp. 197–206.
- Voß, T., N. Hansen, and C. Igel. “Improved step size adaptation for the MO-CMA-ES.” In *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference* (2010), pp. 487–494.
- Wills, A.G. and T.B. Schön. “On the construction of probabilistic Newton-type algorithms.” In *Proceedings of the 56th IEEE Annual Conference on Decision and Control* (2017), pp. 6499–6504.
- Zenios, S.A. *Practical Financial Optimization: Decision Making for Financial Engineers*. Blackwell: Malden (2007).
- Zhang, R., N. Langrené, Y. Tian, Z. Zhu, F. Klebaner, and K. Hamza. “Dynamic portfolio optimization with liquidity cost and market impact: a simulation-and-regression approach.” *Quantitative Finance*, Vol. 19, No. 3 (2018), pp. 519–532.