



Portfolio construction as linearly constrained separable optimization

Nicholas Moehle^{1,2,3} · Jack Gindi^{1,2,3} · Stephen Boyd^{1,2,3} · Mykel J. Kochenderfer^{1,2,3}

Received: 17 August 2021 / Revised: 29 June 2022 / Accepted: 29 June 2022 /
Published online: 11 November 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Mean–variance portfolio optimization problems often involve separable nonconvex terms, including penalties on capital gains, integer share constraints, and minimum nonzero position and trade sizes. We propose a heuristic algorithm for such problems based on the alternating direction method of multipliers (ADMM). This method allows for solve times in tens to hundreds of milliseconds with around 1000 securities and 100 risk factors. We also obtain a bound on the achievable performance. Our heuristic and bound are both derived from similar results for other optimization problems with a separable objective and affine equality constraints. We discuss a concrete implementation in the case where the separable terms in the objective are piecewise quadratic, and we empirically demonstrate its effectiveness for tax-aware portfolio construction.

Keywords Convex optimization · portfolio optimization · proximal methods · convex envelope

✉ Nicholas Moehle
nicholasmoehle@gmail.com

Jack Gindi
gindije@gmail.com

Stephen Boyd
boyd@stanford.edu

Mykel J. Kochenderfer
mykel@stanford.edu

¹ BlackRock AI Labs, 820 Ramona St, Palo Alto, CA 94301, USA

² Department of Electrical Engineering, Stanford University, 450 Serra Mall, Stanford, CA 94305, USA

³ Department of Aeronautics and Astronautics, Stanford University, 496 Lomita Mall, Stanford, CA 94305, USA

1 Introduction

The mean–variance portfolio optimization problem of Markowitz (1952) has a quadratic objective and linear equality constraints, allowing for a simple analytical solution. The problem can be extended by including position limits or a long-only constraint (Markowitz 1955; Sharpe 1963; Grinold and Kahn 1999). Although the resulting problem no longer has an analytical solution, it can be efficiently solved as a quadratic program (Boyd and Vandenberghe 2004, pp. 55–156). Many extensions have been proposed that take into account investment restrictions, risk or leverage limits, and trading costs (Boyd et al. 2017). When these extensions preserve the convexity of the portfolio optimization problem, the extended problems can be solved quickly and reliably (Boyd and Vandenberghe 2004, Sect. 1.3.1).

Some important portfolio optimization problems involve nonconvex constraints and objective terms. For example, we may require that asset holdings be in integral numbers of shares, or we may want to penalize realized capital gains (Moehle et al. 2021) or the number of securities traded (Lobo et al. 2007). These extensions are often irrelevant or negligible for large institutional accounts, but they can be important for small accounts such as those arising in separately managed account (SMA) platforms (Benidis et al. 2018). Some of these nonconvex portfolio optimization problems can be reformulated as mixed-integer convex optimization problems (Mansini et al. 2015) and solved exactly using commercial or open source solvers. Such solvers are often fast, but occasionally have very long solve times, often hundreds of times more than for similar convex problems.

This article focuses on nonconvex portfolio optimization problems with nonconvex penalties and constraints that separate across assets. This problem is a special case of the *separable–affine problem*, *i.e.*, the problem of minimizing a separable objective function with affine equality constraints. We propose a heuristic algorithm for solving such problems based on the alternating direction method of multipliers (ADMM). This method solves problems to moderate accuracy quickly, even when the separable functions are very complicated. This speed is obtained because each separable function is interfaced only through its proximal operator, which only involves a few arithmetic operations. This separable–affine form also makes it easy to compute a lower bound on the problem value by replacing the separable functions by their convex envelopes and solving the resulting convex optimization problem. We give a fast algorithm for constructing these convex envelopes for piecewise quadratic functions, which often appear in portfolio optimization.

This article is structured as follows. In Sect. 2, we introduce the portfolio optimization problem with separable nonconvex holding and trading costs. In Sect. 3, we introduce the separable–affine problem and show how the portfolio optimization problem can be expressed in this form. In Sect. 4, we discuss lower bounds on the optimal value of the separable–affine problem. In Sect. 5 we show how to solve separable–affine problems using ADMM. In Sect. 6, we discuss implementation details when the separable functions are piecewise quadratic. Section 7 concludes with an application to tax-aware portfolio construction.

2 Portfolio optimization

2.1 Basic portfolio optimization problem

We first present mean–variance portfolio optimization with separable holding and trading costs. The problem is to decide how much to invest in each of l assets. This decision is represented by $h \in \mathbf{R}^l$, where h_i is the fraction of the portfolio value to be invested in asset i .

We choose h by solving the optimization problem

$$\begin{aligned} &\text{maximize } \alpha^T h - \gamma^{\text{risk}} h^T V h - \gamma^{\text{trd}} \phi^{\text{trd}}(h - h^{\text{init}}) - \gamma^{\text{hld}} \phi^{\text{hld}}(h) \\ &\text{subject to } \eta^{\text{lb}} \leq \mathbf{1}^T h \leq \eta^{\text{ub}}. \end{aligned} \tag{1}$$

The objective of (1) trades off expected return, risk, trading costs, and holding costs, with positive tradeoff parameters γ^{risk} , γ^{trd} , and γ^{hld} . The vector $\alpha \in \mathbf{R}^l$ is the expected return forecast for the n assets, meaning $\alpha^T h$ is the expected portfolio return. The matrix $V \in \mathbf{S}_{++}^l$ (the set of symmetric positive definite $l \times l$ matrices) is the asset return covariance matrix, making $h^T V h$ the variance of the portfolio return. We assume V has the traditional factor model form

$$V = X \Sigma X^T + D, \tag{2}$$

where $X \in \mathbf{R}^{n \times k}$ is the factor exposure matrix, $\Sigma \in \mathbf{S}_{++}^k$ is the factor covariance matrix, and $D \in \mathbf{S}_{++}^n$ is the diagonal matrix of idiosyncratic variances with $D_{ii} > 0$ (Grinold and Kahn 1999; Boyd et al. 2017).

The constraint specifies that the fraction of the account value that is invested (given by $\mathbf{1}^T h$) is between η^{lb} and η^{ub} . Setting $\eta^{\text{lb}} = 0.98$ and $\eta^{\text{ub}} = 0.99$, for example, means that between 98 and 99% of the account value must be invested after the trade is carried out, leaving 1–2% in cash.

The trading cost $\phi^{\text{trd}}(h - h^{\text{init}})$ is the cost of trading from the initial portfolio h^{init} to h , and $\phi^{\text{hld}}(h)$ is the cost of holding portfolio h . We do *not* assume these two functions are convex, but we do assume they are separable, *i.e.*,

$$\phi^{\text{trd}}(u) = \sum_{i=1}^l \phi_i^{\text{trd}}(u_i), \quad \phi^{\text{hld}}(h) = \sum_{i=1}^l \phi_i^{\text{hld}}(h_i).$$

Because ϕ^{trd} and ϕ^{hld} are not convex, problem (1) is difficult to solve exactly in general. One way to solve it is to reformulate it as a mixed-integer convex problem, typically a mixed-integer quadratic program or a mixed-integer second-order cone program, both of which can be solved using standard methods. In practice, these methods often solve problem instances quickly, but sometimes the solution times can be extremely long.

In this paper, we propose a different approach, which exploits the fact that the nonconvex terms are separable. Problems with separable nonconvex terms have been studied extensively, and many effective solution methods have been proposed for them. We discuss problems of this type in more detail in Sect. 3.

2.2 Examples of separable trading costs

We give some examples of trading cost functions ϕ^{trd} . Although we discuss each individually, in practice these functions would be combined together into a single composite trading cost function.

Transaction cost The traditional transaction cost model is

$$\phi_i^{\text{trd}}(u) = s_i |u| + d_i |u|^{3/2}.$$

The first term models the cost of crossing the bid–ask spread, where $s_i \geq 0$ is one-half the bid–ask spread of asset i , and $|\cdot|$ denotes the element-wise absolute value. The second term models the cost of price impact, where $d_i \geq 0$ is the market impact parameter for asset i . The $3/2$ power is applied elementwise.

Minimum trade size A minimum trade size is a constraint that for each asset i , we either do not trade it, or we trade at least u_i^{min} of it. In this case, we have

$$\phi_i^{\text{trd}}(u_i) = \begin{cases} 0 & \text{if } |u_i| \geq u_i^{\text{min}} \text{ or } u_i = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Per-trade cost. A per-trade cost has the form

$$\phi_i^{\text{trd}}(u_i) = \begin{cases} 0 & \text{if } u_i = 0 \\ c_i^{\text{trd}} & \text{otherwise,} \end{cases}$$

where c_i^{trd} is the cost of trading asset i . If these costs are the same for all assets, the portfolio-level trading cost is proportional to the cardinality $\mathbf{card}(u)$, *i.e.*, number of nonzero elements in u . This can be used to model a per-trade commission levied on all trades.

Tax liability Let $L : \mathbf{R}^l \rightarrow \mathbf{R}$ denote the tax liability function, such that $L(h - h^{\text{init}})$ is the immediate tax liability incurred from realized capital gains by trading into portfolio h . This function is separable across the assets, *i.e.*, it has the form

$$\phi_i^{\text{trd}}(u_i) = L_i(u_i),$$

where $L_i(u_i)$ is the tax liability from trading asset i . It is also piecewise affine and has domain $[-h_i^{\text{init}}, \infty)$. An explicit description of L_i is complicated (Moehle et al. 2021, Sect. 3). (The function L_i is especially complicated when the asset i has been traded within the past 30 days because of the wash sale rule, but it is always a piecewise affine function.)

2.3 Examples of separable holding costs

In this section, we provide some examples of trading cost functions ϕ^{hld} . Although we discuss each individually, they are combined into a composite holding cost function.

Position limits Asset-level position limits have the form

$$\phi_i^{\text{hld}}(h_i) = \begin{cases} 0 & \text{if } h_i^{\text{lb}} \leq h_i \leq h_i^{\text{ub}} \\ \infty & \text{otherwise,} \end{cases} \tag{3}$$

where limits for asset i satisfy $h_i^{\text{lb}} \leq h_i^{\text{ub}}$.

Minimum holding size A minimum holding size is a constraint that for each asset i , we either do not hold it, or we hold at least h_i^{min} of it. In this case, we have

$$\phi_i^{\text{hld}}(h_i) = \begin{cases} 0 & \text{if } |h_i| \geq h_i^{\text{min}} \text{ or } h_i = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Per-asset holding cost A per-asset holding cost has the form

$$\phi_i^{\text{hld}}(h_i) = \begin{cases} 0 & \text{if } h_i = 0 \\ c_i^{\text{hld}} & \text{otherwise,} \end{cases} \tag{4}$$

where c_i^{hld} is the cost of holding asset i . If these costs are the same for all assets, the portfolio-level holding cost is proportional to the cardinality $\text{card}(h)$, *i.e.*, the number of nonzero elements in h . Penalizing this value models the account overhead associated with maintaining a portfolio with many assets.

Integer share constraint We can restrict the portfolio to hold an integer number of shares of each asset, *i.e.*, $h_i/p_i \in \mathbf{Z}$, where p_i is the per-share price of asset i . In this case, we have

$$\phi_i^{\text{hld}}(h_i) = \begin{cases} 0 & \text{if } h_i/p_i \in \mathbf{Z} \\ \infty & \text{otherwise.} \end{cases}$$

3 Separable–affine problem

We propose to solve the portfolio optimization problem (1) as a *separable–affine problem* (SAP). In this section, we introduce the SAP problem and discusses some of its properties.

Definition The separable–affine problem is

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n f_i(x_i) \\ & \text{subject to } Ax = b, \end{aligned} \tag{5}$$

with variable $x \in \mathbf{R}^n$. The parameters are $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$, as well as the separable functions $f_i : \mathbf{R} \rightarrow \mathbf{R} \cup \{\infty\}$. The (separable) objective is $f(x) = \sum_{i=1}^n f_i(x_i)$,

We use infinite values of f_i to encode constraints, and define $\text{dom } f_i = \{x_i \mid f_i(x_i) < \infty\}$ and $\text{dom } f = \text{dom } f_1 \times \dots \times \text{dom } f_n$. We will assume that for each i ,

dom f_i is a non-empty union of a finite number of intervals. We also assume each f_i is closed. We say that x is feasible if $x \in \mathbf{dom} f$ and $Ax = b$, and define the (equality constraint) residual associated with x as $r = Ax - b$. We denote the optimal value of the SAP (5) as p^* and a solution (if one exists) as x^* .

Scaling We observe here for future use that both the variables and constraints in the SAP (5) can be scaled, yielding another (equivalent) SAP. Let $E \in \mathbf{R}^{n \times n}$ be diagonal and invertible, and $D \in \mathbf{R}^{m \times m}$ be invertible. With the change of variable $\tilde{x} = E^{-1}x$, and scaling the equality constraints by the matrix D , we obtain the problem

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n f_i(E_{ii}\tilde{x}_i) \\ & \text{subject to } DAE\tilde{x} = Db, \end{aligned} \tag{6}$$

with variable \tilde{x} , which is also an SAP with data

$$\tilde{A} = DAE, \quad \tilde{b} = Db, \quad \tilde{f}_i(\tilde{x}_i) = f_i(E_{ii}\tilde{x}_i), \quad i = 1, \dots, n.$$

From a solution \tilde{x}^* of this problem, we can recover a solution of the original SAP as $x^* = E\tilde{x}^*$.

3.1 Portfolio optimization as a SAP

To express problem (1) as a separable–affine problem, we introduce two additional variables: the cash fraction $c = 1 - \mathbf{1}^T h$ and the factor exposure vector $y = C^T Xh$, where C is a Cholesky factor of Σ , i.e., $CC^T = \Sigma$. Problem (1) is then

$$\begin{aligned} & \text{maximize } \alpha^T h - \gamma^{\text{risk}}(y^T y + h^T Dh) - \gamma^{\text{trd}}\phi^{\text{trd}}(h - h^{\text{init}}) - \gamma^{\text{hld}}\phi^{\text{hld}}(h) \\ & \text{subject to } y = C^T Xh \\ & \quad c + \mathbf{1}^T h = 1 \\ & \quad 1 - \eta^{\text{ub}} \leq c \leq 1 - \eta^{\text{lb}}. \end{aligned} \tag{7}$$

The variables are $h \in \mathbf{R}^l$, $c \in \mathbf{R}$, and $y \in \mathbf{R}^k$. Minimizing the negative of the objective yields a separable–affine problem; the exact values of the parameters f , A , and b are given in Appendix A.

3.2 Solving SAPs

Convex case If the functions f_i are all convex, the SAP (5) is a convex optimization problem that is readily solved.

Exhaustive search If the number of degrees of freedom $n - m$ is very small (say, no more than 3 or 4), we can solve the SAP by exhaustive search, which involves evaluating $f(x)$ on a grid of points in the subspace $\{x \in \mathbf{R}^n \mid Ax = b\}$.

Divide and conquer When m is very small (say, no more than 3), we can use variations on dynamic programming or divide and conquer to solve the SAP. For a subset $S \subseteq$

$\{1, \dots, n\}$, we define the value function

$$V_S(z) = \inf \left\{ \sum_{i \in S} f_i(x_i) \mid \sum_{i \in S} x_i a_i = z \right\},$$

where a_i are the columns of A . When m is very small, we can represent V_S by its values on a grid, or by the coefficients in a suitable basis.

We have $V_{\{k\}}(z) = f_k(x_k^*)$ if $x_k^* a_k = z$ for some x_k^* , and $V_{\{k\}}(z) = \infty$ otherwise. We also have $V_{\{1, \dots, n\}}(b) = p^*$, as well as the dynamic programming property

$$V_{S \cup T}(z) = \inf_{u, v} \{V_S(u) + V_T(v) \mid u + v = z\},$$

for disjoint S and T . When m is very small, we can form $V_{S \cup T}$ by brute force search over a grid of values u in \mathbf{R}^m . Thus, we can form $V_{S \cup T}$, given V_S and V_T (modulo our gridding or basis approximation).

This suggests the following divide and conquer method. We start with the collection of n value functions V_S with S a singleton (which are the same as the functions f_i). We then combine pairs, using the combining formula above, to obtain around n value functions V_S , with $|S| = 2$. We continue this way around $\log_2 n$ times to obtain $V_{\{1, \dots, n\}}$. Evaluating this function at $z = b$ gives p^* .

Examining the portfolio problem (1) (or its separable–affine form (7)), note that $m = k + 1$, where k is the number of factors in the risk model. This implies that portfolio optimization problems with separable nonconvexities can be solved exactly if the number of risk factors is small.

The general case is hard SAP includes mixed-integer linear programs as a special case, which in turn includes as a further special case the 3-SAT problem, which is NP-complete. Thus, we do not expect to develop a global optimization method that is efficient in the worst case; we only hope for efficiency of a global method on many (or even just some) problem instances that arise in practice.

3.3 Special cases

The SAP includes a number of well-known problem classes as special cases.

LP and MILP With $f_i(x_i) = c_i x_i + I_+(x_i)$, where I_+ is the indicator function of \mathbf{R}_+ ,

$$I_+(u) = \begin{cases} 0 & \text{if } u \geq 0 \\ \infty & \text{if } u < 0, \end{cases}$$

the SAP reduces to the generic linear programming (LP) problem. Adding the indicator function of $\{0, 1\}$ to f_i we obtain a general mixed-integer linear program (MILP).

Indefinite quadratic programming. The SAP generalizes the indefinite quadratic programming problem (IQP)

$$\begin{aligned} & \text{minimize } x^T P x + q^T x \\ & \text{subject to } A x = b \\ & \quad x \geq 0, \end{aligned} \tag{8}$$

where P is a symmetric matrix. We factor P as $P = F D F^T$, where D is $r \times r$ diagonal with nonzero diagonal entries and $F \in \mathbf{R}^{n \times r}$. This can be obtained from an eigendecomposition of P . Thus, we have $x^T P x = \sum_{i=1}^n D_{ii} z_i^2$ for $z = F^T x$. Adding this new variable, we obtain the SAP

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n (q_i(x_i) + I_+(x_i)) + \sum_{i=1}^r D_{ii} z_i^2 \\ & \text{subject to } \begin{bmatrix} A & 0 \\ F^T & -I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \end{aligned}$$

with variable (x, z) . This is equivalent to the IQP above. Adding the indicator function of $\{0, 1\}$ to each f_i yields a general mixed-integer IQP.

Limitations The feasible set of a SAP consists of the Cartesian product of unions of intervals (i.e., $\text{dom } f_i$) and the affine set $\{x \mid Ax = b\}$. The feasible set of an SAP is therefore the union of a finite number of convex polyhedra. This observations tells us that optimization problems with non-polyhedral feasible sets, such as problems with quadratic constraints, cannot be exactly represented as SAPs.

4 Bounds

In this section, we describe a lower bound on p^* obtained by solving a convex relaxation of (5).

Convex envelope The convex envelope of f is

$$f^{**}(x) = \sup\{g(x) \mid g \leq f, g \text{ convex}\},$$

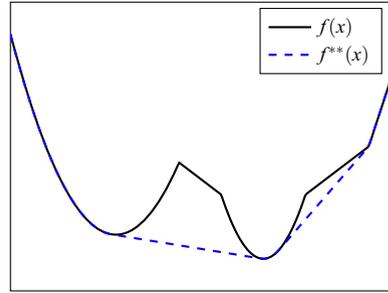
i.e., its value at a point x is the greatest value of $g(x)$ obtained for any convex function g that minorizes f . The convex envelope is also the (Fenchel) conjugate of the conjugate of f , i.e., $(f^*)^*$, where the superscript $*$ denotes conjugation (Rockafellar 1970, Sect. 5). (This explains why we denote the convex envelope of f as f^{**} .) If f is convex, then we have $f^{**} = f$. An example is shown in Fig. 1.

Bound from convex relaxation. By replacing the separable functions f_i in problem (5) with their convex envelopes f_i^{**} , we obtain the problem

$$\begin{aligned} & \text{minimize } \sum_{i=1}^n f_i^{**}(x_i) \\ & \text{subject to } A x = b, \end{aligned} \tag{9}$$

which we call the *convex relaxation* of (5). The relaxed problem (9) is a convex separable–affine problem. The optimal value of this relaxation is denoted d^* . (The

Fig. 1 A nonconvex function f (solid black line) and its convex envelope f^{**} (dashed blue line). (Color figure online)



reason for this notation will become clear later.) Because $f_i^{**} \leq f_i$, we have $d^* \leq p^*$. Indeed, if f_i is convex for all i , then $f_i = f_i^{**}$ and therefore $d^* = p^*$.
Dual problem The dual problem of (5) is

$$\begin{aligned} &\text{maximize } \lambda^T b - \sum_{i=1}^n f_i^*(-v_i) \\ &\text{subject to } A^T \lambda = v. \end{aligned} \tag{10}$$

The variables are $v \in \mathbf{R}^n$ and $\lambda \in \mathbf{R}^m$. Because the linear term in (10) is separable, the dual problem is itself a separable–affine problem.

The dual problem is always a convex optimization problem, even when the primal problem (5) is not. The optimal value of the dual problem is d^* , which is the optimal value of the relaxed problem (9). Weak duality, which always holds, states that $d^* \leq p^*$. If (5) is convex, then strong duality holds, which means that $d^* = p^*$ (Bertsekas 1999, Prop. 5.2.1).

5 ADMM

We can apply the alternating direction method of multipliers (ADMM) to the separable–affine problem (5).

ADMM-form problem We start with the equivalent problem

$$\begin{aligned} &\text{minimize } \mathcal{I}_{\mathcal{A}}(z) + \sum_{i=1}^n f_i(x_i) \\ &\text{subject to } x = z \end{aligned} \tag{11}$$

with variables $x \in \mathbf{R}^n$ and $z \in \mathbf{R}^n$. Here $\mathcal{I}_{\mathcal{A}}$ is the indicator function over the affine constraints of (5), *i.e.*,

$$\mathcal{I}_{\mathcal{A}}(z) = \begin{cases} 0 & \text{if } Az = b \\ \infty & \text{otherwise.} \end{cases}$$

Augmented Lagrangian The augmented Lagrangian of (11) is

$$L(x, z, \lambda) = f(x) + \mathcal{I}_{\mathcal{A}}(z) + \frac{1}{2} \|x - z + \lambda\|^2.$$

Our definition of the augmented Lagrangian lacks the parameter $\rho > 0$ from the standard definition given by Bertsekas (1999). Including this parameter is equivalent to using problem scaling parameter $D = \rho I$, as discussed in Sect. 3.

ADMM iterations The ADMM algorithm iterates x^k, z^k , and λ^k , for $k = 0, 1, 2, \dots$, are

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L(x, z^k, \lambda^k) \\ z^{k+1} &= \operatorname{argmin}_z L(x^{k+1}, z, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + x^{k+1} - z^{k+1}. \end{aligned} \tag{12}$$

The initial values are z^0 and λ^0 .

Convergence If f is convex and a solution to (5) exists, then as $k \rightarrow \infty$, we have $f(x^k) \rightarrow p^*$, $Ax^k \rightarrow b$, and $\lambda^k \rightarrow \lambda^*$, where λ^* is an optimal dual variable to (5). In the general case when f is nonconvex, there is no such guarantee.

Solving the linear system Here we describe the z^{k+1} update in (12). Minimizing the augmented Lagrangian involves solving the equality-constrained least-squares problem

$$\begin{aligned} &\text{minimize } \|z - x^{k+1} + \lambda^k\|^2 \\ &\text{subject to } Az = b \end{aligned}$$

with variable z . The minimizing z (which becomes z^{k+1}) can be found by solving the linear system of equations

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} z^{k+1} \\ \nu \end{bmatrix} = \begin{bmatrix} x^k - \lambda^k \\ b \end{bmatrix}. \tag{13}$$

For each iteration in the ADMM algorithm, we solve (13) for different values of the right-hand side (*i.e.*, for different values of x^k and λ^k .) We can do this efficiently by factorizing this matrix before the first iteration and caching it for repeated use (Stellato et al. 2020, Sect. 3.1).

Separable update The update rule for x^{k+1} in (12) can be written

$$x^{k+1} = \operatorname{argmin}_x \left(f(x) + \frac{1}{2} \|x - z^k + \lambda^k\|^2 \right).$$

Because f is separable, we can perform this update by solving n univariate optimization problems. The update for each component x_i is given by

$$x_i^{k+1} = \operatorname{argmin}_{x_i} \left(f_i(x_i) + \frac{1}{2} (x_i - z_i^k + \lambda_i^k)^2 \right).$$

These updates can be expressed using the proximal operator of f_i :

$$x_i^{k+1} = \mathbf{prox}_{f_i}(z_i^k - \lambda_i^k).$$

(For more information, see Boyd et al. (2011).) Several methods can be used to solve these small problems, including exhaustive search. For the portfolio problem (1), f_i is piecewise quadratic, and the proximal operator can be readily computed, as shown in Sect. 6.2.

Initialization If f is convex, the ADMM iterates converge to a solution of (5) regardless of the initialization. However, if f is not convex, the choice of initialization can have a large impact on the result of ADMM.

In this case, one good choice is to initialize the iterates using the solution to the relaxed problem (9). This has the added benefit of providing a lower bound on the optimal problem value, which can be used to judge the quality of the iterates produced by ADMM.

Scaling Although the scaling parameters D and E do not change the solution of (5), they can have a large impact on the rate of convergence of ADMM. (When (5) is not convex, they can also affect the quality of the iterates.) One effective method to choose D and E is through a simple hyperparameter search, in which we tune D and E to minimize the run time over a sampling of similar problems. This requires expressing D and E in terms of a small number of free parameters, as in done for example in Sect. 7.1. Another approach is to choose D and E to reduce the condition number of A , e.g., through equilibration (Stellato et al. 2020, Sect. 5).

Termination criteria We need termination criteria that work well in the case when f is not convex. Termination criteria for ADMM, when applied to convex problems, are discussed in (Boyd et al. 2011, Sect. 3.3).

For any candidate point $x \in \mathbf{R}^n$ satisfying $Ax = b$, we define the pseudo-objective and residual values as

$$\begin{aligned} o(x) &= f(\Pi_{\text{dom}(f)}(x)) \\ r(x) &= \mathbf{dist}(x, \text{dom}(f)). \end{aligned}$$

We use the following procedure to check for termination:

1. From the current iterates, find a point x satisfying $Ax = b$.
2. Compute $o(x)$ and $r(x)$.
3. If $r(x) < \epsilon^{\text{res}}$ and $o(x) < o^{\text{best}}$, then update o^{best} and the best iterates.
4. Terminate if the best objective o^{best} has not improved by more than ϵ^{obj} in more than N iterations.

There are several ways to carry out step 1. For example, we can take the iterate x^k , which satisfies $Ax^k = b$. For the portfolio optimization problem (1), we can take the elements of the iterate z^k that represent the asset holdings h and compute the corresponding cash amount c and factor exposures y . Then the vector $x = (h, c, y)$ satisfies $Ax = b$.

6 Implementation

Carrying out ADMM requires evaluating the proximal operator of univariate functions, and computing the bound from 4 requires computing the convex envelope. In this

section, we show how to carry out these two operations for the specific case in which the univariate functions are piecewise quadratic. We use the notation φ for a generic univariate function, which stands in for f_i in (5).

6.1 Piecewise quadratic functions

Consider the piecewise quadratic function φ with k pieces, defined as

$$\varphi(x) = \begin{cases} p_1x^2 + q_1x + r_1 & \text{if } x \in [a_1, b_1] \\ \vdots & \vdots \\ p_kx^2 + q_kx + r_k & \text{if } x \in [a_k, b_k] \\ +\infty & \text{otherwise} \end{cases} \quad (14)$$

with $a_1 \leq b_1 \leq a_2 \leq \dots \leq a_k \leq b_k$. We denote by φ_i the i th piece of φ , *i.e.*,

$$\varphi_i(x) = \begin{cases} p_ix^2 + q_ix + r_i & \text{if } x \in [a_i, b_i] \\ +\infty & \text{otherwise.} \end{cases} \quad (15)$$

Because we do not require φ to be continuous, and each piece is defined over a closed interval, φ may be multiply defined for some values of x (at the boundaries of some intervals). To remedy this, we use the convention that $\varphi(x)$ is the minimum over all such candidate values, *i.e.*, $\varphi(x) = \min\{\varphi_i(x) \mid \forall i \text{ s.t. } x \in [a_i, b_i]\}$. We also assume the description of φ is irreducible, *i.e.*, none of the k pieces can be dropped without changing the function value at some point.

6.2 Proximal operator

The proximal operator of φ is

$$\mathbf{prox}_\varphi(u) = \operatorname{argmin}_x \left(\varphi(x) + \frac{1}{2}(x - u)^2 \right). \quad (16)$$

Evaluating the proximal operator is done in two steps. First, we compute the piecewise quadratic function $\varphi(x) + (x - u)^2/2$, which is done by adding $1/2$, $-u$, and $u^2/2$ to the coefficients p_i , q_i , and r_i , for all i . Computing the minimizer of this function can be done by computing the minimum value of each piece, taking the minimum over these values, and then finding a value of x that attains this minimum.

If φ is convex, its proximal operator of φ can be expressed as

$$\mathbf{prox}_\varphi(u) = \begin{cases} a_1 & \text{if } u \in (-\infty, (2p_1 + 1)a_1 + q_1] \\ (u - q_1)/(1 + 2p_1) & \text{if } u \in [(2p_1 + 1)a_1 + q_1, (2p_1 + 1)b_1 + q_1] \\ \vdots & \vdots \\ a_j & \text{if } u \in [(2p_{j-1} + 1)b_{j-1} + q_{j-1}, (2p_j + 1)a_j + q_j] \\ (u - q_j)/(1 + 2p_j) & \text{if } u \in [(2p_j + 1)a_j + q_j, (2p_j + 1)b_j + q_j] \\ \vdots & \vdots \\ a_k & \text{if } u \in [(2p_{k-1} + 1)b_{k-1} + q_{k-1}, (2p_k + 1)a_k + q_k] \\ (u - q_k)/(1 + 2p_k) & \text{if } u \in [(2p_k + 1)a_k + q_k, (2p_k + 1)b_k + q_k] \\ b_k & \text{if } u \in [(2p_k + 1)b_k + q_k, \infty). \end{cases}$$

We note that some of these intervals may be degenerate.

6.3 Convex envelope

In this section we show how to compute the convex envelope of φ , which is required for computing the bound discussed in Sect. 4. This follows the same lines as Gardiner and Lucet (2010). To do this, first note that the convex envelope φ^{**} can be computed recursively:

$$\begin{aligned} \varphi^{**} &= \min\{\psi^i, \varphi_i\}^{**} \\ \psi^i &= \min\{\psi^{i-1}, \varphi_{i-1}\}^{**} \\ &\vdots \\ \psi^2 &= \min\{\psi^1, \varphi_1\}^{**} \\ \psi^1 &= \varphi_1. \end{aligned} \tag{17}$$

Here \min is the pointwise minimum operation between functions. An example of this recursion for the function

$$\varphi(x) = \begin{cases} x^2 - 3x - 3 & \text{if } x \in [-\infty, 3] \\ -x + 3 & \text{if } x \in [3, 4] \\ 2x^2 - 20x + 47 & \text{if } x \in [4, 6] \\ x - 7x & \text{if } x \in [6, 7.5] \\ 4x + 29 & \text{if } x \in [7.5, \infty], \end{cases} \tag{18}$$

is shown in Fig. 2.

We now discuss how to carry out each line in the recursion (17), *i.e.*, how to compute ψ^{i+1} given ψ^i . Note that ψ^i and φ_i are convex functions. The graph of ψ^i lies to the

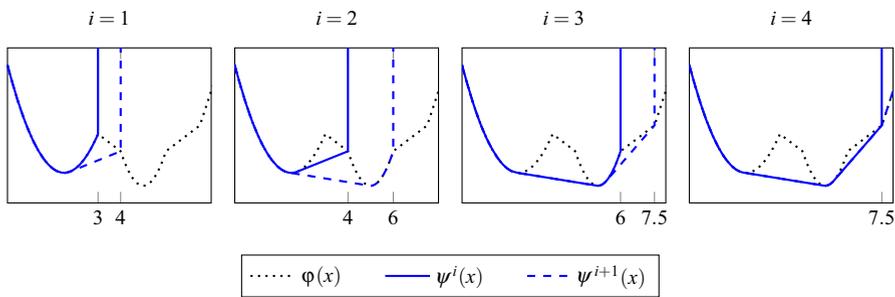
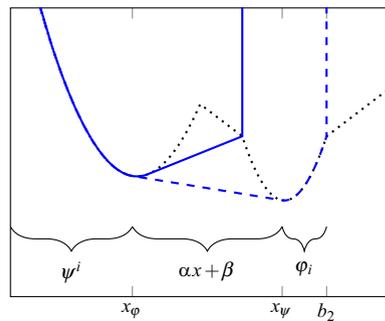


Fig. 2 Recursive computation of the convex envelope for the function shown in Fig. 1

Fig. 3 The three components that make up ψ^{i+1} according to Eq. (20), with φ given in (18) and $i = 2$



left of the graph of φ_i , meaning that $x \leq z$ for all $x \in \text{dom}\psi^i$ and $z \in \text{dom}\varphi_i$. Therefore ψ^{i+1} has the simple form

$$\psi^{i+1}(x) = \begin{cases} \psi^i(x) & \text{if } x \in [a_1, x_\psi] \\ \alpha x + \beta & \text{if } x \in [x_\psi, x_\varphi] \\ \varphi_i(x) & \text{if } x \in [x_\varphi, b_i] \end{cases} \tag{19}$$

for some $\alpha \in \mathbf{R}$, $\beta \in \mathbf{R}$, $x_\varphi \in \text{dom}\varphi_i$, and $x_\psi \in \text{dom}\psi^i$. This is shown in Fig. 3. We allow for the case when $x_\varphi = -\infty$ or $x_\psi = \infty$; in these cases, the first or last interval is degenerate, and can be ignored. The parameters α and β are unique; x_φ and x_ψ need not be. Finding these parameters is straightforward but tedious; further details are given in Appendix B.

7 Numerical example

In this section we provide an example of the method applied to tax-aware portfolio optimization with small account sizes. We use a passive, index-tracking strategy with $\alpha = 2\gamma^{\text{risk}} V h^{\text{bm}}$, where the elements of $h^{\text{bm}} \in \mathbf{R}^l$ are the weights of a benchmark portfolio. This means that minimizing the objective term $\alpha^T h + \gamma^{\text{risk}} h^T V h$ is equivalent to minimizing the (squared) active risk $(h - h^{\text{bm}})^T V (h - h^{\text{bm}})$.

The trading cost combines the bid–ask spread model with a penalty on the number of trades and capital gains:

$$\phi^{\text{trd}}(u) = \gamma^{\text{sprd}} s^T |u| + c^{\text{trd}} \mathbf{card}(u) + \gamma^{\text{tax}} L(u).$$

The holding cost combines the asset position limits (3) and the per-asset holding cost (4). We use the lower bound $h^{\text{lb}} = 0$, which encodes a long-only constraint, and the upper bound for asset i is $h_i^{\text{ub}} = \max\{3h_i^{\text{bm}}, h_i^{\text{init}}\}$, i.e., it is the greater of the current holdings and 3 times the benchmark weight. The combined holding cost is

$$\phi^{\text{hld}}(h) = \begin{cases} c^{\text{hld}} \mathbf{card}(h) & \text{if } 0 \leq h \leq h^{\text{ub}} \\ \infty & \text{otherwise.} \end{cases}$$

7.1 Problem data

Backtest setup. We generate instances of problem (1) from backtests of a tax loss harvesting strategy. Our backtest dataset consists of 204 months over a 17 year period from August 2002 through August 2019. We use this dataset to carry out 12 staggered six-year-long backtests, with the first starting in August 2002 and ending in July 2008, and the the last starting in August 2013 and ends in July 2019. Every month, a single instance of problem (1) is solved to rebalance the portfolio. We then save this problem instance, which we use to evaluate our heuristic and bound. The full details of this setup are described by Moehle et al. (2021), Sect. 6, including the sources of data used and the exact timing of the rebalance trades. (The main difference between our formulation and theirs is the addition of the per-asset holding and trading cost terms.)

When solving problem (1), we use the parameters

$$\gamma^{\text{risk}} = 100, \quad c^{\text{hld}} = c^{\text{trd}} = 3 \times 10^{-5}, \quad \gamma^{\text{tax}} = 1.$$

The parameters γ^{trd} and γ^{hld} are redundant, and were set to 1. With these values, the portfolio maintains active risk around 0.5–1%, holds only 200–300 securities of the S&P 500, and typically trades around 30 securities per month. The invested fraction is maintained between $\eta^{\text{lb}} = 0.98$ and $\eta^{\text{ub}} = 0.99$.

Generated problems The procedure given above resulted in 692 instances of problem (1). The mean optimal utility U^* across these problems ranged from –1028 to 184 basis points (0.0001, one hundredth of one percent) with a mean and standard deviation of 94 basis points and 57 basis points, respectively.

Algorithm parameters We used the scaling parameters

$$D = \mathbf{diag}(100\mathbf{1}_l, 3, 100\mathbf{1}_k), \quad E = \mathbf{diag}(100\mathbf{1}_m),$$

where $\mathbf{1}_l$ and $\mathbf{1}_k$ are the l and k dimensional vectors with all entries one. We used the stopping criterion parameters

$$\epsilon^{\text{res}} = 3 \times 10^{-4}, \quad \epsilon^{\text{obj}} = 10^{-5}.$$

Fig. 4 Distribution of suboptimality gaps $p_{\text{admm}} - d^*$

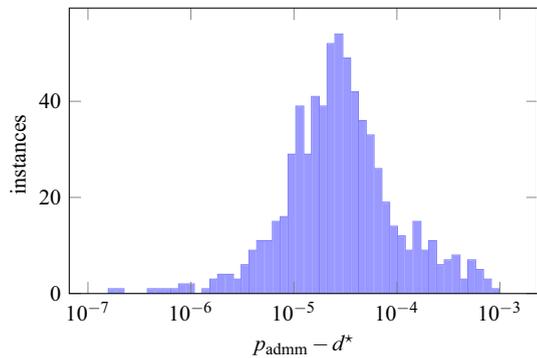
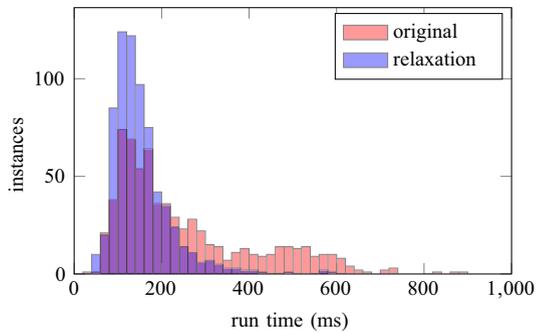


Fig. 5 Run time distributions for the ADMM algorithm applied to the original problem (1) as well as the relaxed problem (9)



We check the termination conditions once every 10 steps, and terminate if the objective have not improved in more than $N = 50$ iterations.

7.2 Results

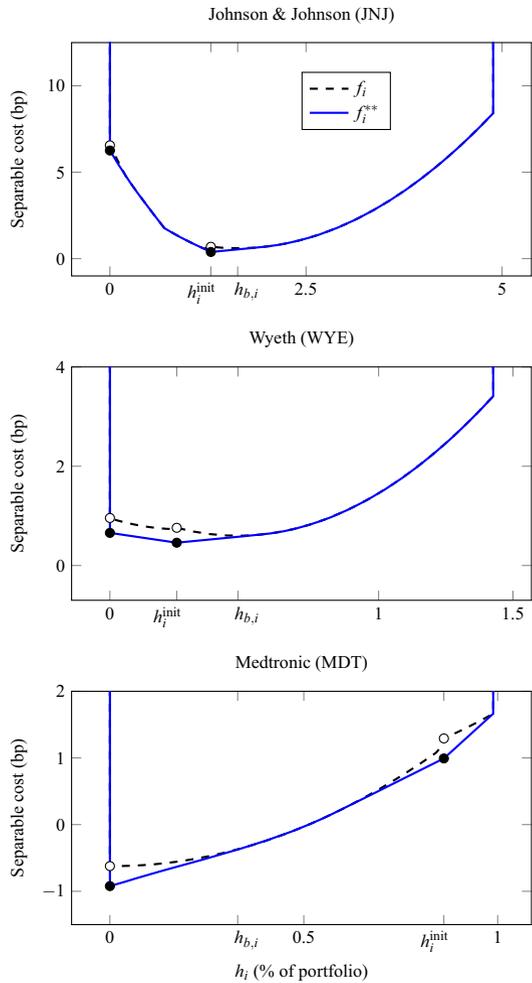
Heuristic quality All of the 692 problems we ran the algorithm on converged. To evaluate the ADMM heuristic, we compare the objective values obtained by ADMM on the 692 problem instances, which we denote p_{admm} , to the lower bounds d^* obtained by solving the convex relaxation (9). The optimal problem value p^* lies between these values, *i.e.*,

$$d^* \leq p^* \leq p_{\text{admm}}.$$

Figure 4 shows the differences $p_{\text{admm}} - d^*$. These range from 0 to 10 basis points, with mean 0.6 basis points and standard deviation 1.1 basis points. These values are quite small compared to the problems values p^* , which range from -1028 to 184 basis points. This implies that the ADMM heuristic produces nearly optimal points on all 692 problem instances.

Speed In addition to being accurate, ADMM was also fast. It solved the original (nonconvex) problems in an average of 251 ms (with standard deviation 159 ms), and solved the relaxed (convex) problems in an average of 152 ms (with a standard

Fig. 6 The cost functions f_i and their relaxations f_i^{**} for the stocks of Johnson & Johnson, Wyeth, and Medtronic, for a single simulated trade



deviation 67 ms). Figure 5 shows the distributions of run times for original and relaxed problems.

7.3 Single problem instance

Figure 6 shows the separable cost functions f_i corresponding to the stocks of Johnson & Johnson (JNJ), Wyeth (WYE), and Medtronic (MDT). We show the initial holdings h_i^{init} , the benchmark holdings h_i^{bm} , and the optimal post-trade holdings h_i^* .

The upward curvature of all three functions is primarily due to the specific risk term $D_{ii}(h_i - h_i^{bm})^2$. Each f_i has two discontinuities: one at $h_i = 0$ that corresponds to the holding cost (4) and one at $h_i = h_i^{init}$ that corresponds to the holding cost (4). Due to the larger scale of f_i for Johnson & Johnson, discontinuities are less severe, relative

to the rest of the function, than they are for the other two assets. In addition to these discontinuities, there is also a nonconvex kink present in all three plots around h_i^{init} , due to the tax liability L_i (Moehle et al. 2021, Sect. 3). Despite these nonconvexities, the separable cost functions f_i are generally well approximated by their convex envelopes f_i^{**} , which helps explain why the upper bound from ADMM and the lower bound (given in Sect. 7.2) are so close.

Appendix A Portfolio construction parameters

We can convert problem (7) into the separable–affine form (5). The variable is $x = (h, c, y) \in \mathbf{R}^{l+k+1}$. The affine constraint parameters are

$$A = \begin{bmatrix} C^T X & 0 & -I \\ \mathbf{1}^T & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The part of the separable function corresponding to asset i is

$$f_i(h_i) = \alpha_i h_i - \gamma^{\text{risk}} D_{ii} h_i^2 + \gamma^{\text{trd}} \phi_i^{\text{trd}}(h_i - h_i^{\text{init}}) + \gamma^{\text{hld}} \phi_i^{\text{hld}}(h_i)$$

for $i = 1, \dots, l$. The component of f corresponding to c is

$$f_{l+1}(c) = \begin{cases} 0 & \text{if } 1 - \eta^{\text{ub}} \leq c \leq 1 - \eta^{\text{lb}} \\ \infty & \text{otherwise.} \end{cases}$$

The component of f corresponding to y is $f_i(y_i) = y_i^2$, for $i = l + 2, \dots, l + k + 1$.

Appendix B Convex envelope details

We can compute the convex envelope of a piecewise quadratic function as discussed in Sect. 6.3. To do this, we must compute the parameters α, β, x_ψ , and x_φ from (19). We do this making use of the following two facts.

The first fact is that candidate values of α, β, x_ψ , and x_φ are valid if and only if the function $\alpha x + \beta$ is tangent to ψ^i at x_ψ and to φ_i at x_φ . This holds if and only if

$$\alpha \in \partial\psi^i(x_\psi), \quad \psi^i(x_\psi) = \alpha x_\psi + \beta, \quad \alpha \in \partial\varphi_i(x_\varphi), \quad \varphi_i(x_\varphi) = \alpha x_\psi + \beta, \quad (20)$$

where $\partial\psi^i$ and $\partial\varphi_i$ are the subdifferentials of ψ^i and φ_i . These conditions can be easily checked given a set of candidate parameters α, β, x_ψ , and x_φ .

The second fact is that (19) can be rewritten

$$\psi^{i+1}(x) = \min\{\psi^i, \varphi_i\}^{**} = \begin{cases} \psi_1^i(x) & \text{if } x \in [\tilde{a}_1, \tilde{b}_1] \\ \vdots & \vdots \\ \psi_{j-1}^i(x) & \text{if } x \in [\tilde{a}_{j-1}, \tilde{b}_{j-1}] \\ \min\{\psi_j^i, \varphi_i\}^{**} & \text{if } x \in [\tilde{a}_j, b_i], \end{cases} \tag{21}$$

where $\tilde{a}_1 \leq \tilde{b}_1 \leq \dots \leq \tilde{a}_j$ are the endpoints of the pieces of ψ^i . In other words, the function ψ^{i+1} matches ψ^i up to the j th piece of ψ^i , and after that, is equal to the convex envelope of the pointwise minimum of ψ_j^i and φ_i . This convex envelope (the last piece in (21)) is easy to compute, as $\min\{\psi_j^i, \varphi_i\}$ is a piecewise quadratic with two pieces. (We discuss how to do this in Sect. 1.)

This means that once j is known, we can use (21) to compute ψ^{i+1} . To find j , we simply try all pieces of ψ^i , compute the right-hand side of (21), and check if the resulting parameters α, β, x_ψ and x_φ satisfy the conditions (20).

Appendix B.1 Piecewise-quadratic functions with two pieces

The function $\min\{\psi_j^i, \varphi_i\}$ in (21) is piecewise quadratic with two pieces. Here we discuss how to compute the convex envelope of such functions.

Let $g(x)$ be a PWQ with two pieces: $g_1(x) = p_1x^2 + q_1x + r_1$ on $[a_1, b_1]$, and $g_2(x) = p_2x^2 + q_2x + r_2$ on $[a_2, b_2]$, with $b_1 \leq a_2$. In this case, g^{**} has the form

$$g^{**}(x) = \begin{cases} g_1(x) & \text{if } x \in [a_1, x_1] \\ h(x) = \alpha x + \beta & \text{if } x \in [x_1, x_2] \\ g_2(x) & \text{if } x \in [x_2, b_2] \end{cases} \tag{22}$$

for some α, β, x_1 , and x_2 . These parameters are real valued, but we allow for the case when $x_1 = -\infty$ or $x_2 = \infty$; in these cases, first or last interval is degenerate, and can be ignored.

Similarly to (20) above, it is necessary and sufficient for the parameters α, β, x_1 , and x_2 to satisfy

$$\alpha \in \partial g_1(x_1), \quad g_1(x_1) = \alpha x_1 + \beta, \quad \alpha \in \partial g_2(x_2), \quad g_2(x_2) = \alpha x_2 + \beta. \tag{23}$$

How these checks are carried out in practice depends on whether x_1 (or x_2) are in the interior or boundary of the domain of g_1 (or g_2), or whether $x_1 = -\infty$ (or $x_2 = \infty$).

Appendix B.1.1 The midpoint-to-midpoint case

We first consider the case when x_1 is in the interior of the domain of g_1 , *i.e.*, $a_1 < x_1 < b_1$, and x_2 is in the interior of the domain of g , *i.e.*, $a_2 < x_2 < b_2$. In this case,

we must have

$$g_1(x_1) = h(x_1), \quad g_2(x_2) = h(x_2), \quad g'_1(x_1) = h'(x_1), \quad g'_2(x_2) = h'(x_2). \quad (24)$$

By plugging in the values of the functions and their derivatives, we obtain

$$\begin{aligned} p_1x_1^2 + q_1x_1 + r_1 &= \alpha x_1 + \beta \\ p_2x_2^2 + q_2x_2 + r_2 &= \alpha x_2 + \beta \\ 2p_1x_1 + q_1 &= \alpha \\ 2p_2x_2 + q_2 &= \alpha. \end{aligned} \quad (25)$$

These four equations can be reduced to a single quadratic equation with a single unknown. This quadratic equation has at most two solutions, each corresponding to a set of candidate values of α , β , x_1 , and x_2 . To see if these candidate values α , β , x_1 , and x_2 parameterize a valid convex envelope of g , we check if our initial assumption, that $a_1 < x_1 < b_1$ and $a_2 < x_2 < b_2$, holds.

Appendix B.1.2 The midpoint-to-endpoint case

Now we consider the case in which x_1 is in the interior of the domain of g_1 , and $x_2 \in \{a_2, b_2\}$. (We note that the case in which $x_1 \in \{a_1, b_1\}$ and x_2 in the interior of the domain of g can be handled similarly, and we do not discuss it further.)

Finite upper bound. We start with the case when $x_2 < \infty$. To do this, we solve a slightly modified set of equations:

$$g_1(x_1) = h(x_1), \quad g'_1(x_1) = h'(x_1), \quad g_2(x_2) = h(x_2). \quad (26)$$

These are the first three equations of (25). As before, they can be reduced to a single quadratic equation with a single unknown. This quadratic equation has at most two solutions, each corresponding to a set of candidate values of α , β , and x_1 .

To see if these candidate values parameterize a valid convex envelope, we verify that x_1 is in the interior of the domain of g_1 , *i.e.*, $a_1 < x_1 < b_1$, and also the second condition of (23). In the degenerate case in which g_2 is defined over a single point, *i.e.*, $a_2 = b_2$, this second condition always holds; in the non-degenerate case $a_2 < b_2$, the condition is equivalent to $g'_2(x_2) = 2p_2x_2 + q_2 \leq \alpha$ if $x_2 = a_2$, and $g'_2(x_2) = 2p_2x_2 + q_2 \geq \alpha$ if $x_2 = b_2$.

Infinite upper bound. Next we consider the case of $x_2 = \infty$, which may occur when $x_2 = b_2 = \infty$, *i.e.*, the domain of g_2 is unbounded. In this case, we modify the last equation in (26), resulting in the equations

$$g_1(x_1) = h(x_1), \quad g'_1(x_1) = h'(x_1), \quad g'_2(\infty) = h'(\infty). \quad (27)$$

The last equation is equivalent to $p_2 = 0$ and $\alpha = q_2$. To find the corresponding candidate values of β , and x_1 , we solve $g_1(x_1) = h(x_1)$ and $g'_1(x_1) = h'(x_1)$. To check

the validity of the candidate values of α , β , x_1 , and x_2 , we check that $a_1 < x_1 < b_1$ and $p_2 = 0$.

Appendix B.1.3 The endpoint-to-endpoint case

Finally, we consider the case in which $x_1 \in \{a_1, b_1\}$ and $x_2 \in \{a_2, b_2\}$.

Finite upper bound. First suppose a_1 , b_1 , a_2 , and b_2 are all finite. Take h to be the line through the points $(x_1, g_1(x_1))$ and $(x_2, g_2(x_2))$, i.e., the parameters are $\alpha = (g_2(x_2) - g_1(x_1))/(x_2 - x_1)$ and $\beta = \alpha x_1 - g_1(x_1)$. Note that if $x_1 = x_2$, then h is ill-defined, and these candidate values of x_1 and x_2 can be skipped.

To verify that these values of α , β , x_1 , and x_2 parameterize a valid convex envelope of f , we check condition (23). Recall that the first condition only need hold if $a_1 < b_1$, and the second if $a_2 < b_2$. For example, if $a_1 = b_1 < a_2 = b_2$, then any combination of $x_1 \in \{a_1, b_1\}$ and $x_2 \in \{a_2, b_2\}$ are immediately valid.

Infinite upper bound. Now we consider the case when $x_2 = b_2 = \infty$. In this case, we require $h'(\infty) = g_2'(\infty)$, i.e., $p_2 = 0$ and $\alpha = q_2$. We then have $\beta = g_1(x_1) - q_2 x_1$. To verify that α , β , x_1 , and x_2 parameterize a valid envelope of f , we need only check the first condition of (23).

References

- Bertsekas D (1999) Nonlinear programming, 3rd edn. Athena Scientific, Nashua
- Benedis K, Feng Y, Palomar D (2018) Optimization methods for financial index tracking: from theory to practice. *Found Trends Optim* 3(3):171–279
- Boyd S, Busseti E, Diamond S, Kahn R, Nystrup P, Speth J (2017) Multi-Period Trading via Convex Optimization. *Found Trends Optim* 3(1):1–76
- Boyd S, Parikh N, Chu E (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
- Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
- Grinold R, Kahn R (1999) Active Portfolio management, 2nd edn. McGraw-Hill, New York
- Gardiner B, Lucet Y (2010) Convex hull algorithms for piecewise linear-quadratic functions in computational convex analysis. *Set-Valued Var Anal* 18(3–4):467–482
- Lobo M, Fazel M, Boyd S (2007) Portfolio optimization with linear and fixed transaction costs. *Ann Oper Res* 152(1):341–365
- Markowitz H (1952) Portfolio selection. *J Finance* 7(1):7791
- Markowitz H (1955) The optimization of a quadratic function subject to linear constraints. RAND Corporation Tech rep, Santa Monica
- Moehle N, Kochenderfer MJ, Boyd S, Ang A (2021) Tax-aware portfolio construction via convex optimization. *J Optim Theory Appl* 189(2):364–383
- Mansini R, Ogryczak W, Speranza MG (2015) Linear and mixed integer programming for portfolio optimization. Springer, Cham
- Rockafellar R (1970) Convex analysis. Princeton University Press, Princeton
- Sharpe W (1963) A simplified model for portfolio analysis. *Manag Sci* 9(2):277293
- Stellato B, Banjac G, Goulart P, Bemporad A, Boyd S (2020) OSQP: an operator splitting solver for quadratic programs. *Math Program Comput* 12:637–672

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.