

Solving High-Order Portfolios via Successive Convex Approximation Algorithms

Rui Zhou¹, *Student Member, IEEE*, and Daniel P. Palomar¹, *Fellow, IEEE*

Abstract—The first moment and second central moments of the portfolio return, a.k.a. mean and variance, have been widely employed to assess the expected profit and risk of the portfolio. Investors pursue higher mean and lower variance when designing the portfolios. The two moments can well describe the distribution of the portfolio return when it follows the Gaussian distribution. However, the real world distribution of assets return is usually asymmetric and heavy-tailed, which is far from being a Gaussian distribution. The asymmetry and the heavy-tailedness are characterized by the third and fourth central moments, i.e., skewness and kurtosis, respectively. Higher skewness and lower kurtosis are preferred to reduce the probability of extreme losses. However, incorporating high-order moments in the portfolio design is very difficult due to their non-convexity and rapidly increasing computational cost with the dimension. In this paper, we propose a very efficient and convergence-provable algorithm framework based on the successive convex approximation (SCA) algorithm to solve high-order portfolios. The efficiency of the proposed algorithm framework is demonstrated by the numerical experiments.

Index Terms—High-order portfolios, skewness, kurtosis, efficient algorithm, successive convex approximation.

I. INTRODUCTION

MODERN portfolio theory has developed rapidly since Harry Markowitz's seminal paper in 1952, which proposed the mean-variance framework to pursue the trade-off between maximizing the portfolio's profit and minimizing its risk [1]. The profit and risk of a portfolio are measured by the mean and variance, i.e., the first moment and the second central moment, of the portfolio return. The mean-variance framework assumes that the investors prefer a quadratic utility or that the returns of assets follow a Gaussian distribution [2].

However, the mean-variance framework is not widely used in the real market investment. One of the main reasons is that returns of assets in real markets are seldom Gaussian distributed. They are usually asymmetric and more likely to contain outliers or exhibit a heavier tail, making the portfolio return also asymmetric and heavy-tailed [3], [4]. Meanwhile, most investors

Manuscript received July 9, 2020; revised December 31, 2020; accepted January 5, 2021. Date of publication January 13, 2021; date of current version February 5, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Monica F. Bugallo. This work was supported by the Hong Kong GRF 16208917, and 16207019 research grants. (Corresponding author: Rui Zhou.)

The authors are with the Electronic and Computer Engineering, Hong Kong University of Science, and Technology (HKUST), Clear Water Bay, Kowloon, Hong Kong (e-mail: rui.zhou@connect.ust.hk; palomar@ust.hk).

Digital Object Identifier 10.1109/TSP.2021.3051369

would be willing to accept lower expected profit and higher volatility in exchange for more positively skewed and less heavy-tailed portfolio return [5]–[7]. This aspiration has been beyond the characterization of the mean-variance framework. Apart from that, the investors might have different tastes in utility functions. Sometimes the shapes of these utility functions can be significantly different from the quadratic one.

To make up the drawbacks of the mean-variance framework, we need to take high-order moments of the portfolio return into consideration. The asymmetry and heavy-tailedness of portfolio return are well captured by its third and fourth central moments, i.e., skewness and kurtosis. A higher skewness usually means that the portfolio return admits a more positively skewed shape, while the lower kurtosis usually corresponds to thinner tail. We can extend the mean-variance framework by directly incorporating the high-order moments to obtain the mean-variance-skewness-kurtosis (MVK) framework, where we shall try to strike a balance between maximizing the mean and skewness (odd moments) while minimizing the variance and kurtosis (even moments) [8]–[10]. Besides, such extension can be seen as approximating a general expected utility function with its Taylor series expansion truncated to the four most important order terms [11]. There also exist some other high-order portfolios within the MVK framework. For example, the MVK tilting portfolios [12] are obtained by “tilting” a given portfolio to the MVK efficient frontier.

Although there are many advantages of the MVK framework, solving such high-order portfolio optimization problems is quite challenging. First, the third and fourth central moments are both non-convex functions, making the problems in general NP-hard [13]. These problems are traditionally solved by some metaheuristic optimization tools, e.g., differential evolution [9] and genetic algorithms [10]. However, they are essentially performing a time-consuming random search [14], [15]. A method based on the Difference of Convex (DC) algorithm was proposed to solve the MVK portfolio problem to a stationary point [8]. It was further improved based on the Difference-of-Convex-Sums-of-Squares (DC-SOS) decomposition techniques [16]. However, the DC method may converge too slowly and it is only applicable to small-size problems. Second, the complexity of computing the value or the gradients of high-order moments grows rapidly with the problem dimension. The classical gradient descent method and backtracking line search also become inapplicable when the problem dimension grows large. Therefore, it is meaningful and necessary to design efficient algorithms for solving high-order portfolios.

To this end, the major goal of this paper is to develop an efficient algorithm framework based on the successive convex approximation (SCA) to solve high-order portfolios. The SCA algorithm solves the original intractable problem by constructing and solving a sequence of strongly convex approximating problems [17]–[20]. In this paper, we propose an easy approach to construct the approximation for the non-convex functions. This allows to construct a sequence of convex problems compatible with existing efficient solvers that can obtain the solutions to the original high-order portfolio optimization problems. The convergence of the proposed algorithm framework to a stationary point is established. In addition, owing to their low computational complexity, the algorithms are amenable for high-dimensional applications. Extensive numerical experiments are performed to corroborate our claims.

The paper is organized as follows. We first give the preliminary knowledge on the high-order moments of portfolio return in Section II and then pose the problem formulations in Section III. The SCA algorithm and its special cases are introduced in Section IV. In Section V and Section VI, we derive our algorithms based on the SCA algorithm to solve the high-order portfolios. The complexity and convergence analysis of the proposed algorithms are discussed in Section VII. In Section VIII, we present some other formulations of high-order portfolio problems and indicate the applicability of our proposed algorithm framework. The numerical experiments are given in Section IX. Finally, the conclusion of this paper is summarized in Section X.

II. PRELIMINARIES: THE MOMENTS OF PORTFOLIO RETURN

Denote by $\mathbf{r} \in \mathbb{R}^N$ the returns of N assets and $\mathbf{w} \in \mathbb{R}^N$ the portfolio weights. The return of this portfolio is $\mathbf{w}^T \mathbf{r}$ with expected value, i.e., first moment,

$$\phi_1(\mathbf{w}) = \mathbb{E}[\mathbf{w}^T \mathbf{r}] = \mathbf{w}^T \boldsymbol{\mu}, \quad (1)$$

where $\boldsymbol{\mu} = \mathbb{E}(\mathbf{r})$ is the mean vector of the assets' returns. Denote by $\tilde{\mathbf{r}} = \mathbf{r} - \boldsymbol{\mu}$ the centered returns, the q -th central moment of the portfolio return is $\mathbb{E}[(\mathbf{w}^T \mathbf{r} - \mathbf{w}^T \boldsymbol{\mu})^q] = \mathbb{E}[(\mathbf{w}^T \tilde{\mathbf{r}})^q]$, which gives us the follows:

- The second central moment, a.k.a. variance, of the portfolio return is

$$\begin{aligned} \phi_2(\mathbf{w}) &= \mathbb{E}[(\mathbf{w}^T \tilde{\mathbf{r}})^2] \\ &= \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w}] \\ &= \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}, \end{aligned} \quad (2)$$

where $\boldsymbol{\Sigma} = \mathbb{E}[\tilde{\mathbf{r}} \tilde{\mathbf{r}}^T]$ is the covariance matrix.

- The third central moment, a.k.a. skewness, of the portfolio return is

$$\begin{aligned} \phi_3(\mathbf{w}) &= \mathbb{E}[(\mathbf{w}^T \tilde{\mathbf{r}})^3] \\ &= \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w}] \\ &= \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} (\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T) (\mathbf{w} \otimes \mathbf{w})] \\ &= \mathbf{w}^T \boldsymbol{\Phi} (\mathbf{w} \otimes \mathbf{w}), \end{aligned} \quad (3)$$

where $\boldsymbol{\Phi} = \mathbb{E}[\tilde{\mathbf{r}} (\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)]$ is the co-skewness matrix.

- The fourth central moment, a.k.a. kurtosis, of the portfolio return is

$$\begin{aligned} \phi_4(\mathbf{w}) &= \mathbb{E}[(\mathbf{w}^T \tilde{\mathbf{r}})^4] \\ &= \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w}] \\ &= \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} (\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T) (\mathbf{w} \otimes \mathbf{w}) \tilde{\mathbf{r}}^T \mathbf{w}] \\ &= \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} (\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T) (\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w})] \\ &= \mathbf{w}^T \boldsymbol{\Psi} (\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w}), \end{aligned} \quad (4)$$

where $\boldsymbol{\Psi} = \mathbb{E}[\tilde{\mathbf{r}} (\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)]$ is the co-kurtosis matrix.

The gradients of $\phi_1(\mathbf{w})$ and $\phi_2(\mathbf{w})$ w.r.t. \mathbf{w} are $\boldsymbol{\mu}$ and $2\boldsymbol{\Sigma}\mathbf{w}$, while their Hessians are $\mathbf{0}$ and $2\boldsymbol{\Sigma}$, respectively. But the gradient and the Hessian of $\phi_3(\mathbf{w})$ and $\phi_4(\mathbf{w})$ are more complicated to derive and we give next some useful results.

Lemma 1: The gradient and Hessian of the skewness and kurtosis are given by:

$$\begin{aligned} \nabla \phi_3(\mathbf{w}) &= 3\boldsymbol{\Phi} (\mathbf{w} \otimes \mathbf{w}), \\ \nabla \phi_4(\mathbf{w}) &= 4\boldsymbol{\Psi} (\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w}), \\ \nabla^2 \phi_3(\mathbf{w}) &= 6\boldsymbol{\Phi} (\mathbf{I} \otimes \mathbf{w}), \\ \nabla^2 \phi_4(\mathbf{w}) &= 12\boldsymbol{\Psi} (\mathbf{I} \otimes \mathbf{w} \otimes \mathbf{w}). \end{aligned} \quad (5)$$

Proof: See Appendix A. ■

Corollary 2: The gradient and Hessian of the skewness and kurtosis admit the following relations:

$$\nabla \phi_3(\mathbf{w}) = \frac{1}{2} \nabla^2 \phi_3(\mathbf{w}) \mathbf{w}, \quad (6)$$

$$\nabla \phi_4(\mathbf{w}) = \frac{1}{3} \nabla^2 \phi_4(\mathbf{w}) \mathbf{w}. \quad (7)$$

Proof: Using Lemma 1, we have $3\phi_3(\mathbf{w}) = \mathbf{w}^T \nabla \phi_3(\mathbf{w})$. Then taking the derivative of both sides w.r.t. \mathbf{w} , we get $3\nabla \phi_3(\mathbf{w}) = \nabla \phi_3(\mathbf{w}) + \nabla^2 \phi_3(\mathbf{w}) \mathbf{w}$, which further leads to (6). Equation (7) can be derived similarly. ■

Note that $\nabla^2 \phi_3(\mathbf{w}) = 6 \sum_{k=1}^N \Phi_{ij}^{(k)} w_k$ and $\nabla^2 \phi_4(\mathbf{w}) = 12 \sum_{k,l=1}^N \Psi_{ij}^{(k,l)} w_k w_l$ can be easily obtained from Lemma 1, where, with some abuse of notation, $\Phi_{ij}^{(k)} = \mathbb{E}[\tilde{r}_i \tilde{r}_j \tilde{r}_k]$ and $\Psi_{ij}^{(k,l)} = \mathbb{E}[\tilde{r}_i \tilde{r}_j \tilde{r}_k \tilde{r}_l]$ are the corresponding elements of matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$.

A high expected value and low variance of the portfolio return are naturally chased by investors to increase the profit and decrease the risk. Besides, in the non-Gaussian case, a high skewness and low kurtosis are also desirable as they can reduce the probability of extreme losses. As shown in Fig. 1, a positively skewed portfolio return is significantly less likely to suffer extreme losses than a negatively skewed one. Besides, we can see from Fig. 2 that a lower kurtosis shows also a thinner tail, which alleviates the appearance of extreme returns. In general, investors have a preference for odd moments while dislike even moments.

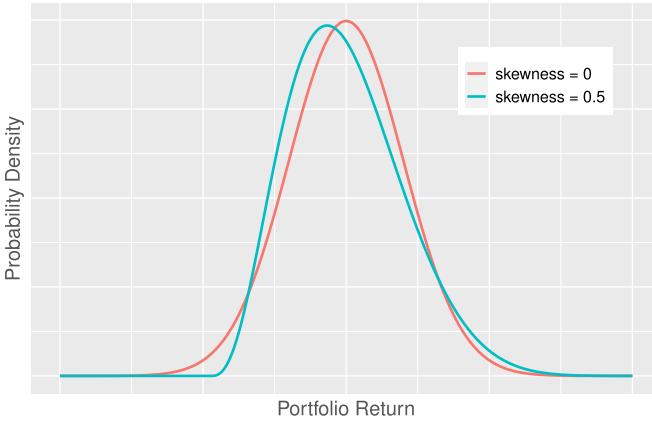


Fig. 1. The implication of skewness.

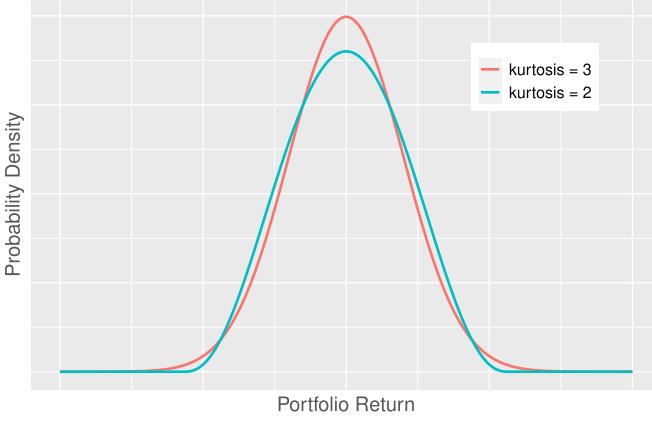


Fig. 2. The implication of kurtosis.

III. PROBLEM FORMULATION

A. MVSK Portfolio

The classical Markowitz's mean-variance (MV) portfolio [1] is obtained by solving the following problem:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} \quad -\mathbf{w}^T \boldsymbol{\mu} + \lambda \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \\ & \text{subject to} \quad \mathbf{w} \in \mathcal{W}, \end{aligned} \quad (8)$$

where $\lambda \geq 0$ is a parameter striking a balance between the expected return (defined by the mean $\mathbf{w}^T \boldsymbol{\mu}$) and the portfolio risk (defined by the variance $\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}$), \mathcal{W} is the feasible set of portfolio weights, which we set as

$$\mathcal{W} = \{\mathbf{w} | \mathbf{1}^T \mathbf{w} = 1, \|\mathbf{w}\|_1 \leq L\}, \quad (9)$$

where $L \geq 1$ is the leverage constraint of the portfolio [21]. Specifically, when $L = 1$, \mathcal{W} reduces to the no shorting constraint: $\{\mathbf{w} | \mathbf{1}^T \mathbf{w} = 1, \mathbf{w} \geq \mathbf{0}\}$. The mean and the variance are actually the first moment and the second central moment of the portfolio return. However, the real world assets return usually appears to be asymmetric and with extreme values, which is beyond the characterization of the first two moments. It is reasonable to consider the third and fourth central moments in the portfolio design. A natural way to incorporate the two higher-order moments is revising the objective of problem (8) to obtain the mean-variance-skewness-kurtosis portfolio design

problem [8]–[10]:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} \quad f(\mathbf{w}) = -\lambda_1 \phi_1(\mathbf{w}) + \lambda_2 \phi_2(\mathbf{w}) \\ & \quad - \lambda_3 \phi_3(\mathbf{w}) + \lambda_4 \phi_4(\mathbf{w}) \\ & \text{subject to} \quad \mathbf{w} \in \mathcal{W}, \end{aligned} \quad (10)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ are the parameters for combining the four moments of the portfolio return.

B. MVSK Tilting Portfolio

Directly solving the problem (10) leads us to the MVSK efficient frontier, where we cannot improve any moment without impairing other moments. However, the investors might want to modify another existing portfolio \mathbf{w}_0 toward a MVSK efficient portfolio. This can be done by tilting these portfolios in a direction that increases their first moment and third central moment and decreases their second and fourth central moments [12], i.e.,

$$\begin{aligned} & \underset{\mathbf{w}, \delta}{\text{maximize}} \quad \delta \\ & \text{subject to} \quad \phi_1(\mathbf{w}) \geq \phi_1(\mathbf{w}_0) + d_1 \delta, \\ & \quad \phi_2(\mathbf{w}) \leq \phi_2(\mathbf{w}_0) - d_2 \delta, \\ & \quad \phi_3(\mathbf{w}) \geq \phi_3(\mathbf{w}_0) + d_3 \delta, \\ & \quad \phi_4(\mathbf{w}) \leq \phi_4(\mathbf{w}_0) - d_4 \delta, \\ & \quad (\mathbf{w} - \mathbf{w}_0)^T \boldsymbol{\Sigma} (\mathbf{w} - \mathbf{w}_0) \leq \kappa^2, \\ & \quad \mathbf{w} \in \mathcal{W}, \delta \geq 0, \end{aligned} \quad (11)$$

where $\mathbf{d} = [d_1, d_2, d_3, d_4] \geq \mathbf{0}$ is the tilting direction, $\phi_i(\mathbf{w}_0), i = 1, 2, 3, 4$ are the moments of \mathbf{w}_0 (starting point) for tilting, κ^2 determines the maximum tracking error volatility of \mathbf{w} with respect to the reference portfolio \mathbf{w}_0 .

C. Difficulty of Solving High-Order Portfolios

The MVSK portfolio optimization problem (10) and MVSK tilting portfolio optimization problems (11) are very difficult to solve for two reasons:

- 1) **Non-convexity:** the third and fourth central moments, i.e., $\phi_3(\mathbf{w})$ and $\phi_4(\mathbf{w})$, are non-convex in \mathbf{w} , making both problems (10) and (11) non-convex problems.
- 2) **Computational complexity:** $\boldsymbol{\Psi}$ is of dimension $N \times N^3$, which means the memory complexity is $\mathcal{O}(N^4)$ and the computational complexity of one single evaluation of the fourth moment is $\mathcal{O}(N^4)$. Lemma 1 shows that the computational complexity for computing the gradient of the fourth central moment is also $\mathcal{O}(N^4)$. Then the general gradient descent method and backtracking line search are inappropriate to the high-order portfolio problem.

Due to the non-convexity, the classical convex optimization methods are not applicable, while the general gradient method is also not applicable due to the expensive cost of gradient computation. It is necessary to design a specific algorithm to efficiently solve high-order portfolios. Such an algorithm should converge fast and avoid evaluating the gradients or value of high-order moments frequently. This paper proposes a very

efficient algorithm framework to solve the high-order portfolio optimization problem based on the SCA algorithm. But before that, some background on the SCA algorithm is due in the next section.

IV. THE SUCCESSIVE CONVEX APPROXIMATION ALGORITHM

The successive convex approximation (SCA) algorithm is a general framework especially designed for solving non-convex optimization problems. Instead of solving the original intractable optimization problem, it resorts to successively solving a sequence of strongly convex approximating problems. The convergence of the SCA algorithm can be guaranteed under mild assumptions.

Specifically, consider a non-convex constrained optimization problem,

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \\ & \text{subject to} \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m, \\ & \quad \mathbf{x} \in \mathcal{K}, \end{aligned} \quad (12)$$

where $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are non-convex functions and \mathcal{K} is a convex set. In order to solve the problem (12), which is directly intractable, we may turn to successively solving a sequence of strongly convex approximating problems. Denote by \mathbf{x}^k the current iterate at k -th iteration, then the SCA algorithm constructs a strongly convex approximating problem for (12) as [19]:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad \tilde{f}(\mathbf{x}; \mathbf{x}^k) \\ & \text{subject to} \quad \tilde{g}_i(\mathbf{x}; \mathbf{x}^k) \leq \eta(\mathbf{x}^k), \quad i = 1, \dots, m, \\ & \quad \|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \beta, \\ & \quad \mathbf{x} \in \mathcal{K}, \end{aligned} \quad (13)$$

where $\tilde{f}(\mathbf{x}; \mathbf{x}^k)$ and $\tilde{g}_i(\mathbf{x}; \mathbf{x}^k)$ are the approximating functions for $f(\mathbf{x})$ and $g_i(\mathbf{x})$ at \mathbf{x}^k , the quantity $\eta(\mathbf{x}^k)$ in the surrogate constraints serves to suitably enlarge the feasible set of the subproblem to ensure it is always nonempty, and β is a user-chosen positive constant. The term $\eta(\mathbf{x}^k)$ is defined as

$$\begin{aligned} \eta(\mathbf{x}^k) & \triangleq (1 - \theta) \max_i \{g_i(\mathbf{x}^k)_+\} \\ & + \theta \min_{\mathbf{x}} \left\{ \max_i \{\tilde{g}_i(\mathbf{x}; \mathbf{x}^k)_+\} \mid \mathbf{x} \in \mathcal{K} \right\}, \end{aligned} \quad (14)$$

with $\theta \in (0, 1)$. The general SCA algorithm generates the sequence $\{\mathbf{x}^k\}$ as

$$\begin{cases} \hat{\mathbf{x}}^{k+1} & \leftarrow \text{solve the problem (13),} \\ \mathbf{x}^{k+1} & = \mathbf{x}^k + \gamma^k (\hat{\mathbf{x}}^{k+1} - \mathbf{x}^k), \end{cases} \quad (15)$$

where at each iteration, the first stage is generating the descent direction $\hat{\mathbf{x}}^{k+1} - \mathbf{x}^k$, and the second stage is updating the variable along the solved descent direction with a step-size γ^k satisfying

$$\lim_{k \rightarrow \infty} \gamma^k = 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma^k = \infty. \quad (16)$$

The generated sequence $\{\mathbf{x}^k\}$ is proven to converge to a generalized stationary point of the original problem (12) under the following mild assumptions [19]:

Assumption 1: Let O_β and O_K be open neighborhoods of $\{\mathbf{x}\} \mid \|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \beta\}$ and \mathcal{K} and such that:

On original problem (12):

A1) \mathcal{K} is an nonempty, closed, and convex set.

A2) $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are continuously differentiable with locally Lipschitz gradients on an open set containing \mathcal{K} .

On surrogate function \tilde{f} :

B1) $\tilde{f}(\mathbf{x}; \mathbf{y})$ is a strongly convex function on O_β for every $\mathbf{y} \in \mathcal{K}$ with modulus of strong convexity $c > 0$ independent of \mathbf{y} ;

B2) $\tilde{f}(\mathbf{x}; \mathbf{y})$ is continuous on $O_\beta \times O_K$;

B3) $\nabla_1 \tilde{f}(\mathbf{x}; \mathbf{y})$ is continuous on $O_\beta \times O_K$;

B4) $\nabla_1 \tilde{f}(\mathbf{y}; \mathbf{y}) = \nabla f(\mathbf{y})$ for every $\mathbf{y} \in \mathcal{K}$;

On surrogate constraint \tilde{g}_i :

C1) $\tilde{g}_i(\mathbf{x}; \mathbf{y})$ is a convex function on O_β for every $\mathbf{y} \in \mathcal{K}$;

C2) $\tilde{g}_i(\mathbf{x}; \mathbf{y})$ is continuous on $\mathbb{R}^N \times O_K$;

C3) $\tilde{g}_i(\mathbf{x}; \mathbf{y}) = g_i(\mathbf{y})$ for every $\mathbf{y} \in \mathcal{K}$;

C4) $\nabla_1 \tilde{g}_i(\mathbf{x}; \mathbf{y})$ is continuous on $O_\beta \times O_K$;

C5) $\nabla_1 \tilde{g}_i(\mathbf{y}; \mathbf{y}) = \nabla f(\mathbf{y})$ for every $\mathbf{y} \in \mathcal{K}$;

where $\nabla_1 \tilde{f}(\mathbf{u}; \mathbf{y})$ and $\nabla_1 \tilde{g}_i(\mathbf{u}; \mathbf{y})$ denote the partial gradient of $\tilde{f}(\mathbf{u}; \mathbf{y})$ and $\tilde{g}_i(\mathbf{u}; \mathbf{y})$ evaluated at \mathbf{u} .

We can simplify the surrogate problem (13) accordingly when the following assumptions are additionally satisfied:

- 1) if \mathcal{K} is bounded, then the constraint $\|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \beta$ can be ignored;
- 2) if $\nabla f(\mathbf{x})$ is Lipschitz continuous on \mathcal{K} and $\tilde{g}_i(\mathbf{x}; \mathbf{x}^k) \geq g_i(\mathbf{x})$ is satisfied for every $\mathbf{x} \in \mathcal{K}$, then the constraint $\|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \beta$ can be ignored and $\eta(\mathbf{x}^k) \equiv 0$ [22];
- 3) if $\nabla f(\mathbf{x})$ is Lipschitz continuous on \mathcal{K} and $\tilde{g}_i(\mathbf{x}; \mathbf{x}^k) = g_i(\mathbf{x})$ is satisfied for every $\mathbf{x} \in \mathcal{K}$, then the algorithm reduces to the vanilla SCA algorithm. The constraint $\|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \beta$ can be ignored and $\eta(\mathbf{x}^k) \equiv 0$ [20];
- 4) if \mathcal{K} is bounded, $\tilde{f}(\mathbf{x}; \mathbf{x}^k) \geq f(\mathbf{x})$ and $\tilde{g}_i(\mathbf{x}; \mathbf{x}^k) = g_i(\mathbf{x})$ are satisfied for every $\mathbf{x} \in \mathcal{K}$, then the algorithm reduces to the classical majorization-minimization (MM) method with convex majorization functions. The constraint $\|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \beta$ can be ignored, $\eta(\mathbf{x}^k) \equiv 0$, and γ^k can be simply fixed to 1 [17], [23].

V. SOLVING THE MVSK PORTFOLIO PROBLEM VIA SCA

In this section, we discuss how to solve the problem (10) via the SCA algorithm. We first investigate the Difference of Convex (DC) programming approach for solving the problem (10) [8], which is actually a special case of the MM algorithm [17]. Inspired by this, we herein propose another MM based algorithm by constructing a sequence of tighter upper bound functions. Thus fewer iterations can be expected. However, we further recognize that the MM algorithm might still be too conservative as it requires constructing a global upper bound for the objective function. Therefore, we further propose a general SCA based algorithm for solving the problem (10), where a strongly convex approximating function is constructed for the objective function.

A. Preliminary Approach: DC Algorithm

A DC approach method was proposed in [8] to solve problem (10) by recognizing that $\nabla^2 f(\mathbf{w})$ has a bounded spectral radius under the bounded feasible set \mathcal{W} .

Algorithm 1: DC method for Problem (10).

- 1: Initialize $\mathbf{w}^0 \in \mathcal{W}$ and compute $\tau_{\text{DC}} \geq \rho(\nabla^2 f(\mathbf{w}))$ as in Lemma 3.
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: Calculate $\nabla f(\mathbf{w}^k)$.
- 4: Solve the problem (19) to update \mathbf{w}^{k+1} .
- 5: Terminate loop if converges.
- 6: **end for**

Lemma 3: [8] Given $\mathbf{w} \geq \mathbf{0}$, $\mathbf{1}^T \mathbf{w} = 1$, we have

$$\begin{aligned} \rho(\nabla^2 f(\mathbf{w})) &\leq 2\lambda_2 \|\Sigma\|_\infty + 6\lambda_3 \max_{1 \leq i \leq N} \sum_{j,k=1}^N |\Phi_{ij}^{(k)}| \\ &\quad + 12\lambda_4 \max_{1 \leq i \leq N} \sum_{j,k,l=1}^N |\Psi_{ij}^{(k,l)}|, \end{aligned} \quad (17)$$

where $\rho(\mathbf{X})$ is the spectral radius of \mathbf{X} .

The bound for $\rho(\nabla^2 f(\mathbf{w}))$ provided in Lemma 3 can be easily extended under the constraints in (9) (where instead of no-shorting $\mathbf{w} \geq \mathbf{0}$ we allow some leverage of L with $\|\mathbf{w}\|_1 \leq L$) to

$$\begin{aligned} \rho(\nabla^2 f(\mathbf{w})) &\leq 2\lambda_2 \|\Sigma\|_\infty + 6\lambda_3 L \max_{1 \leq i \leq N} \sum_{j,k=1}^N |\Phi_{ij}^{(k)}| \\ &\quad + 12\lambda_4 L^2 \max_{1 \leq i \leq N} \sum_{j,k,l=1}^N |\Psi_{ij}^{(k,l)}|. \end{aligned}$$

Then we can represent $f(\mathbf{w})$ as

$$f(\mathbf{w}) = \frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w} - \left(\frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w} - f(\mathbf{w}) \right), \quad (18)$$

where both $\frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w}$ and $\frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w} - f(\mathbf{w})$ are convex functions in \mathbf{w} if $\tau_{\text{DC}} \geq \rho(\nabla^2 f(\mathbf{w}))$. Then the classical concave-convex procedure (CCCP) can be employed here by iteratively linearizing the second (concave) term, i.e.,

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \quad \frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w} - \mathbf{w}^T (\tau_{\text{DC}} \mathbf{w}^k - \nabla f(\mathbf{w}^k)) \\ &\text{subject to} \quad \mathbf{w} \in \mathcal{W}, \end{aligned} \quad (19)$$

where $\nabla f(\mathbf{w}^k) = -\lambda_1 \nabla \phi_1(\mathbf{w}^k) + \lambda_2 \nabla \phi_2(\mathbf{w}^k) - \lambda_3 \nabla \phi_3(\mathbf{w}^k) + \lambda_4 \nabla \phi_4(\mathbf{w}^k)$. It is already a convex problem and can be easily solved. Furthermore, we can rewrite it as a convex quadratic programming (QP) problem by introducing a variable $\mathbf{u} \in \mathbb{R}^N$:

$$\begin{aligned} &\underset{\mathbf{w}, \mathbf{u}}{\text{minimize}} \quad \frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w} - \mathbf{w}^T (\tau_{\text{DC}} \mathbf{w}^k - \nabla f(\mathbf{w}^k)) \\ &\text{subject to} \quad \mathbf{1}^T \mathbf{w} = 1, -\mathbf{u} \leq \mathbf{w} \leq \mathbf{u}, \mathbf{1}^T \mathbf{u} \leq L, \end{aligned} \quad (20)$$

which can be very efficiently solved by a QP solver. In the rest of the paper, we will always use this trick to transform the ℓ_1 -norm constraint to linear inequality constraints. The complete DC algorithm for solving the problem (10) is given in Algorithm 1.

B. Preliminary Approach: MM Algorithm

The DC algorithm is a special case of the more general MM algorithm, which works by solving a sequence of global upper bound problems of the original problem [17], [24]. Inspired by the DC approach discussed in the above section, we propose a tighter upper bound function for $f(\mathbf{w})$. Note that the objective in the surrogate problem (19) can be rewritten as

$$\begin{aligned} &\frac{\tau_{\text{DC}}}{2} \mathbf{w}^T \mathbf{w} - \tau_{\text{DC}} (\mathbf{w}^k)^T \mathbf{w} + \nabla f(\mathbf{w}^k)^T \mathbf{w} + \text{const.} \\ &= f(\mathbf{w}^k) + \nabla f(\mathbf{w}^k)^T (\mathbf{w} - \mathbf{w}^k) + \frac{\tau_{\text{DC}}}{2} \|\mathbf{w} - \mathbf{w}^k\|_2^2, \end{aligned} \quad (21)$$

It is actually a global upper bound function of $f(\mathbf{w})$ [17] at \mathbf{w}^k . However, denoting $f(\mathbf{w}) = f_{\text{cvx}}(\mathbf{w}) + f_{\text{ncvx}}(\mathbf{w})$ with $f_{\text{cvx}}(\mathbf{w}) = -\lambda_1 \phi_1(\mathbf{w}) + \lambda_2 \phi_2(\mathbf{w})$ and $f_{\text{ncvx}}(\mathbf{w}) = -\lambda_3 \phi_3(\mathbf{w}) + \lambda_4 \phi_4(\mathbf{w})$, we find $f_{\text{cvx}}(\mathbf{w})$ is already a convex function. Then we can merely construct the upper bound function for $f_{\text{ncvx}}(\mathbf{w})$. Inspired by Lemma 3, we propose a tighter bound for $\rho(\nabla^2 f_{\text{ncvx}}(\mathbf{w}))$ as follows.

Lemma 4: Under the constraints in (9), we have

$$\begin{aligned} \rho(\nabla^2 f_{\text{ncvx}}(\mathbf{w})) &\leq 6\lambda_3 L \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k \leq N} |\Phi_{ij}^{(k)}| \\ &\quad + 12\lambda_4 L^2 \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k, l \leq N} |\Psi_{ij}^{(k,l)}|. \end{aligned} \quad (22)$$

Proof: See Appendix B. ■

Then we can construct, compared with the upper bound function actually used in DC method, a much tighter upper bound function $\check{f}_{\text{ncvx}}(\mathbf{w})$ for $f_{\text{ncvx}}(\mathbf{w})$ at \mathbf{w}^k as [17]:

$$\begin{aligned} \check{f}_{\text{ncvx}}(\mathbf{w}, \mathbf{w}^k) &= f_{\text{ncvx}}(\mathbf{w}^k) + \nabla f_{\text{ncvx}}(\mathbf{w}^k)^T (\mathbf{w} - \mathbf{w}^k) \\ &\quad + \frac{\tau_{\text{MM}}}{2} \|\mathbf{w} - \mathbf{w}^k\|_2^2, \end{aligned} \quad (23)$$

where $\nabla f_{\text{ncvx}}(\mathbf{w}^k) = -\lambda_3 \nabla \phi_3(\mathbf{w}^k) + \lambda_4 \nabla \phi_4(\mathbf{w}^k)$ and $\tau_{\text{MM}} \geq \rho(\nabla^2 f_{\text{ncvx}}(\mathbf{w}))$ can be calculated via Lemma 4. Then a tighter global upper bound function can be constructed for $f(\mathbf{w})$ as $\check{f}(\mathbf{w}, \mathbf{w}^k) = f_{\text{cvx}}(\mathbf{w}) + \check{f}_{\text{ncvx}}(\mathbf{w}, \mathbf{w}^k)$. At each iteration of the MM algorithm, we need solve the following surrogate problem:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \quad \mathbf{w}^T \check{\mathbf{Q}}^k \mathbf{w} + \mathbf{w}^T \check{\mathbf{q}}^k \\ &\text{subject to} \quad \mathbf{w} \in \mathcal{W}, \end{aligned} \quad (24)$$

where $\check{\mathbf{Q}}^k = \lambda_2 \Sigma + \frac{\tau_{\text{MM}}}{2} \mathbf{I}$ and $\check{\mathbf{q}}^k = -\lambda_1 \mu + \nabla f_{\text{ncvx}}(\mathbf{w}^k) - \tau_{\text{MM}} \mathbf{w}^k$. It is a strongly convex QP problem and can be very efficiently solved by a QP solver. The complete MM algorithm for solving the problem (10) is given in Algorithm 2. Compared with the original DC algorithm, the MM algorithm does not introduce any additional computation, while we can expect faster convergence (as is verified in the numerical experiments of Section IX-A).

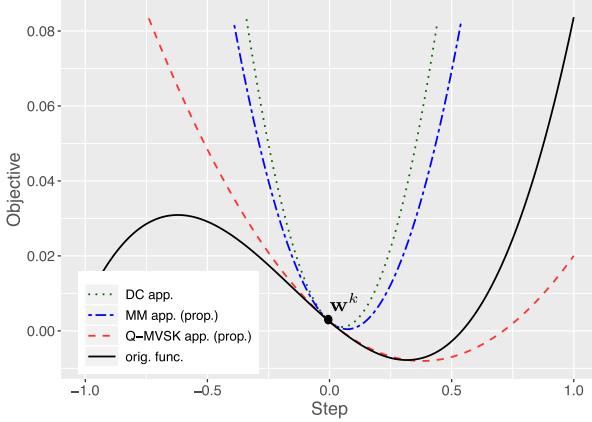


Fig. 3. Illustration of approximating functions.

Algorithm 2: MM Method for Problem (10).

- 1: Initialize $\mathbf{w}^0 \in \mathcal{W}$ and compute $\tau_{\text{MM}} \geq \rho(\nabla^2 f_{\text{ncvx}}(\mathbf{w}))$ as in Lemma 4.
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: Calculate $\nabla f_{\text{ncvx}}(\mathbf{w}^k)$.
- 4: Solve the problem (24) to update \mathbf{w}^{k+1} .
- 5: Terminate loop if converges.
- 6: **end for**

C. Q-MVSK Algorithm

The MM-type methods require constructing a global upper bound approximation, which is sometimes criticized to be too conservative to capture the global landscape of the objective function [18]. Therefore, in this section, we capitalize on the SCA framework and propose the Q-MVSK algorithm to solve the problem (10) via a strongly convex approximation (need not be a global upper bound) for the objective. More specifically, we still leave the convex part $f_{\text{cvx}}(\mathbf{w})$ untouched but construct a second-order approximation for $f_{\text{ncvx}}(\mathbf{w})$ as

$$\begin{aligned} \tilde{f}_{\text{ncvx}}(\mathbf{w}, \mathbf{w}^k) &= f_{\text{ncvx}}(\mathbf{w}^k) + \nabla f_{\text{ncvx}}(\mathbf{w}^k)^T (\mathbf{w} - \mathbf{w}^k) \\ &+ \frac{1}{2} (\mathbf{w} - \mathbf{w}^k)^T \mathbf{H}_{\text{ncvx}}^k (\mathbf{w} - \mathbf{w}^k) + \frac{\tau_{\mathbf{w}}}{2} \|\mathbf{w} - \mathbf{w}^k\|_2^2, \end{aligned} \quad (25)$$

where $\mathbf{H}_{\text{ncvx}}^k$ is an approximation of $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k)$ with $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k) = -\lambda_3 \nabla^2 \phi_3(\mathbf{w}^k) + \lambda_4 \nabla^2 \phi_4(\mathbf{w}^k)$ from Lemma 1, and $\tau_{\mathbf{w}} \geq 0$ is to preserve the strong convexity of $f_{\text{ncvx}}(\mathbf{w}, \mathbf{w}^k)$. Note that $\tau_{\mathbf{w}}$ can be set to 0 when $\lambda_2 > 0$. $\mathbf{H}_{\text{ncvx}}^k$ is a positive semidefinite matrix close to $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k)$ obtained as follows.

Lemma 5: [25] The nearest symmetric positive semidefinite matrix in the Frobenius norm to a real symmetric real matrix \mathbf{X} is $\mathbf{UDiag}(\mathbf{d}_+) \mathbf{U}^T$, where $\mathbf{UDiag}(\mathbf{d}) \mathbf{U}^T$ is the eigenvalue decomposition of \mathbf{X} .

Then we have an approximating function for $f(\mathbf{w})$ as $\tilde{f}(\mathbf{w}, \mathbf{w}^k) = f_{\text{cvx}}(\mathbf{w}) + \tilde{f}_{\text{ncvx}}(\mathbf{w}, \mathbf{w}^k)$. In Fig. 3, the three approximating functions are illustrated along a line on \mathcal{W} . We can see that $\tilde{f}(\mathbf{w}, \mathbf{w}^k)$ can best describe the global behaviour of $f(\mathbf{w})$. At each iteration of the SCA algorithm, we need solve

Algorithm 3: Q-MVSK Algorithm for Problem (10).

- 1: Initialize $\mathbf{w}^0 \in \mathcal{W}$ and pick a sequence $\{\gamma^k\}$.
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: Calculate $\nabla f_{\text{ncvx}}(\mathbf{w}^k)$, $\mathbf{H}_{\text{ncvx}}^k$.
- 4: Solve the problem (26) to obtain $\hat{\mathbf{w}}^{k+1}$.
- 5: $\mathbf{w}^{k+1} = \mathbf{w}^k + \gamma^k (\hat{\mathbf{w}}^{k+1} - \mathbf{w}^k)$.
- 6: Terminate loop if converges.
- 7: **end for**

the following surrogate problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^T \tilde{\mathbf{Q}}^k \mathbf{w} + \mathbf{w}^T \tilde{\mathbf{q}}^k \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \end{aligned} \quad (26)$$

where $\tilde{\mathbf{Q}}^k = \lambda_2 \Sigma + \frac{1}{2} \mathbf{H}_{\text{ncvx}}^k + \frac{\tau_{\mathbf{w}}}{2} \mathbf{I}$ and $\tilde{\mathbf{q}}^k = -\lambda_1 \mu + \nabla f_{\text{ncvx}}(\mathbf{w}^k) - \mathbf{H}_{\text{ncvx}}^k \mathbf{w}^k - \tau_{\mathbf{w}} \mathbf{w}^k$. It is a strongly convex QP problem and can be very efficiently solved by a QP solver. The complete Q-MVSK algorithm for solving the problem (10) is given in Algorithm 3.

VI. SOLVING THE MVSK TILTING PORTFOLIO PROBLEM VIA SCA

In this section, we discuss how to solve the MVSK tilting problem (11), which we rewrite as

$$\begin{aligned} \underset{\mathbf{w}, \delta}{\text{minimize}} \quad & -\delta \\ \text{subject to} \quad & g_i(\mathbf{w}, \delta) \leq 0, \quad i = 1, \dots, 5 \\ & \mathbf{w} \in \mathcal{W}, \delta \geq 0, \end{aligned} \quad (27)$$

where

$$\begin{aligned} g_1(\mathbf{w}, \delta) &= \phi_1(\mathbf{w}_0) - \phi_1(\mathbf{w}) + d_1 \delta, \\ g_2(\mathbf{w}, \delta) &= \phi_2(\mathbf{w}) - \phi_2(\mathbf{w}_0) + d_2 \delta, \\ g_3(\mathbf{w}, \delta) &= \phi_3(\mathbf{w}_0) - \phi_3(\mathbf{w}) + d_3 \delta, \\ g_4(\mathbf{w}, \delta) &= \phi_4(\mathbf{w}) - \phi_4(\mathbf{w}_0) + d_4 \delta, \\ g_5(\mathbf{w}, \delta) &= (\mathbf{w} - \mathbf{w}_{\text{ref}})^T \Sigma (\mathbf{w} - \mathbf{w}_{\text{ref}}) - \kappa^2. \end{aligned} \quad (28)$$

Note that $g_i(\mathbf{w}, \delta)$, $i = 1, 2, 5$ are all convex functions, while $g_i(\mathbf{w}, \delta)$, $i = 3, 4$ are both non-convex functions. We will next explore several options to deal with problem (27), which contains non-convex constraints.

The classical way for solving such constrained problem is the interior-point method (IPM), a.k.a. barrier method, which adds the indicator functions for the inequality constraints to the objective and approximates them with logarithmic barrier functions [26]. The IPM method can be employed to the problem (11) and transform it to

$$\begin{aligned} \underset{\mathbf{w}, \delta}{\text{minimize}} \quad & -t\delta - \sum_{i=1}^5 \log(-g_i(\mathbf{w}, \delta)) \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \delta \geq 0, \end{aligned} \quad (29)$$

where $t > 0$ is a parameter that sets the accuracy of the barrier approximation. Then we could solve the problem (29) via a general

gradient descend method or SCA algorithm. However, due to the implicit constraint $g_i(\mathbf{w}, \delta) \leq 0$, a line search is compulsory at each iteration to guarantee a feasible update of (\mathbf{w}, δ) . As we have discussed before, the computational complexity of a single evaluation of $g_4(\mathbf{w}, \delta)$ is $\mathcal{O}(N^4)$. Then the line search is too computationally expensive to be practical in this problem.

Another way to solve problem (27) could be by constructing a global upper bound approximation for all the non-convex constraints and solve a sequence of inner convex approximating problems. Using the upper bound construction procedure in Section V-B, we can easily construct an inner convex approximating problem for problem (27) at \mathbf{w}^k as:

$$\begin{aligned} & \underset{\mathbf{w}, \delta}{\text{minimize}} \quad -\delta + \frac{\tau_\delta}{2}(\delta - \delta^k)^2 + \frac{\tau_{\mathbf{w}}}{2}\|\mathbf{w} - \mathbf{w}^k\|_2^2 \\ & \text{subject to} \quad g_i(\mathbf{w}, \delta) \leq 0, \quad i = 1, 2, 5, \\ & \quad \check{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \leq 0, \quad j = 3, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \delta \geq 0, \end{aligned} \quad (30)$$

where $\check{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k)$ is the global upper bound of $g_j(\mathbf{w}, \delta)$ at (\mathbf{w}^k, δ^k) , which can be constructed as in Section V-B. The problem (30) is a convex quadratically constrained quadratic programing (QCQP) problem and can be solved via several solvers. However, we can observe from Fig. 3 and the numerical experiments in Section IX-A that such upper bound is very loose and the convergence is slow.

Instead, we propose constructing convex approximations (although not upper bounds) for the non-convex constraints in the following.

A. Preliminary Approach: L-MVSKT Algorithm

The most classical choice, as mentioned in [19], is approximating the objective function by a quadratic function while linearizing all constraints. Therefore, we herein propose the L-MVSKT algorithm by linearizing all the non-linear constraints in problem (11), i.e., the surrogate problem is

$$\begin{aligned} & \underset{\mathbf{w}, \delta}{\text{minimize}} \quad -\delta + \frac{\tau_\delta}{2}(\delta - \delta^k)^2 + \frac{\tau_{\mathbf{w}}}{2}\|\mathbf{w} - \mathbf{w}^k\|_2^2 \\ & \text{subject to} \quad g_1(\mathbf{w}, \delta) \leq 0 \\ & \quad \bar{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \leq \eta(\mathbf{w}^k, \delta^k), \quad j = 2, 3, 4, 5 \\ & \quad \mathbf{w} \in \mathcal{W}, \delta \geq 0, \end{aligned} \quad (31)$$

where $\bar{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k)$ is the linear approximation of $g_j(\mathbf{w}, \delta)$ at (\mathbf{w}^k, δ^k) with

$$\begin{aligned} & \bar{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \\ & = g_1(\mathbf{w}^k, \delta^k) + \nabla_{\mathbf{w}} g_j(\mathbf{w}^k, \delta^k)^T (\mathbf{w} - \mathbf{w}^k) \\ & \quad + \nabla_{\delta} g_j(\mathbf{w}^k, \delta^k)^T (\delta - \delta^k), \quad j = 2, 3, 4. \end{aligned} \quad (32)$$

Besides, $\eta(\mathbf{w}^k, \delta^k)$ here can be computed as

$$\begin{aligned} & \eta(\mathbf{w}^k, \delta^k) \\ & \triangleq (1 - \theta) \max_{j=2,3,4,5} \left\{ g_j(\mathbf{w}^k, \delta^k)_+ \right\} \\ & \quad + \theta \min_{\mathbf{w}, \delta} \left\{ \max_{j=2,3,4,5} \left\{ \bar{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k)_+ \right\} \mid (\mathbf{w}, \delta) \in \bar{\mathcal{W}} \right\}, \end{aligned} \quad (33)$$

Algorithm 4: L-MVSKT Algorithm for Problem (11).

- 1: Initialize $\mathbf{w}^0 \in \mathcal{W}$ and pick $\tau_\delta, \tau_{\mathbf{w}}$ and a sequence $\{\gamma^k\}$.
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: Calculate $\nabla \phi_3(\mathbf{w}^k), \nabla \phi_4(\mathbf{w}^k)$.
- 4: Solve problem (35) and compute $\eta(\mathbf{w}^k, \delta^k)$ as in (33).
- 5: Solve problem (31) to obtain $\hat{\mathbf{w}}^{k+1}$.
- 6: $\mathbf{w}^{k+1} = \mathbf{w}^k + \gamma^k(\hat{\mathbf{w}}^{k+1} - \mathbf{w}^k)$.
- 7: Terminate loop if converges.
- 8: **end for**

where $\bar{\mathcal{W}}$ is a convex set defined as

$$\bar{\mathcal{W}} = \{(\mathbf{w}, \delta) \mid \mathbf{w} \in \mathcal{W}, g_1(\mathbf{w}, \delta) \leq 0, \delta \geq 0\}. \quad (34)$$

The second term in equation (33) is obtained as t from solving the following problem:

$$\begin{aligned} & \underset{\mathbf{w}, \delta, t}{\text{minimize}} \quad t \\ & \text{subject to} \quad \bar{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \leq t, \quad j = 2, 3, 4, 5, \\ & \quad (\mathbf{w}, \delta) \in \bar{\mathcal{W}}, t \geq 0. \end{aligned} \quad (35)$$

Problem (31) is a convex QP problem and problem (35) is a linear programing (LP) problem. Both of them can be very efficiently solved by a QP solver and a LP solver, respectively. The complete L-MVSKT algorithm is given in the Algorithm 4.

B. Q-MVSKT Algorithm

In the above section, we have proposed the L-MVSKT algorithm for solving the MVSK tilting problem (11). However, it requires us to linearize the tractable convex quadratic constraints and the simple linearization is rarely regarded as a proper approximation for non-convex constraints. In Section V-C, we have proposed a quadratic approximation for the third and fourth central moments. It shows great advantages from the numerical experiments presented in Section IX-A. Therefore, similar to Section V-C, we can construct a quadratic approximation for the non-convex constraints in problem (11) while not approximating the already convex constraints, i.e.,

$$\begin{aligned} & \underset{\mathbf{w}, \delta}{\text{minimize}} \quad -\delta + \frac{\tau_\delta}{2}(\delta - \delta^k)^2 + \frac{\tau_{\mathbf{w}}}{2}\|\mathbf{w} - \mathbf{w}^k\|_2^2 \\ & \text{subject to} \quad g_i(\mathbf{w}, \delta) \leq 0, \quad i = 1, 2, 5, \\ & \quad \tilde{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \leq \eta(\mathbf{w}^k, \delta^k), \quad j = 3, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \delta \geq 0. \end{aligned} \quad (36)$$

Here $\tilde{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k)$ is the quadratic approximating function of $g_j(\mathbf{w}, \delta)$ at (\mathbf{w}^k, δ^k) :

$$\begin{aligned} & \tilde{g}_3(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \\ & = \phi_3(\mathbf{w}_0) - \phi_3(\mathbf{w}^k) + d_3\delta - \nabla \phi_3(\mathbf{w}^k)^T (\mathbf{w} - \mathbf{w}^k) \\ & \quad + \frac{1}{2}(\mathbf{w} - \mathbf{w}^k)^T \mathbf{H}_\Phi^k (\mathbf{w} - \mathbf{w}^k), \\ & \tilde{g}_4(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \end{aligned}$$

Algorithm 5: Q-MVSKT Algorithm for Problem (11).

```

1: Initialize  $\mathbf{w}^0 \in \mathcal{W}$  and pick  $\tau_\delta$ ,  $\tau_w$  and a sequence
    $\{\gamma^k\}$ .
2: for  $k = 0, 1, 2, \dots$  do
3:   Calculate  $\nabla\phi_3(\mathbf{w}^k)$ ,  $\nabla\phi_4(\mathbf{w}^k)$ ,  $\mathbf{H}_\Phi^k$ , and  $\mathbf{H}_\Psi^k$ .
4:   Solve problem (40) and compute  $\eta(\mathbf{w}^k, \delta^k)$  as in
   (38).
5:   Solve problem (36) to obtain  $\hat{\mathbf{w}}^{k+1}$ .
6:    $\mathbf{w}^{k+1} = \mathbf{w}^k + \gamma^k(\hat{\mathbf{w}}^{k+1} - \mathbf{w}^k)$ .
7:   Terminate loop if converges.
8: end for

```

$$\begin{aligned}
&= \phi_4(\mathbf{w}^k) - \phi_4(\mathbf{w}_0) + d_4\delta + \nabla\phi_4(\mathbf{w}^k)^T(\mathbf{w} - \mathbf{w}^k) \\
&\quad + \frac{1}{2}(\mathbf{w} - \mathbf{w}^k)^T \mathbf{H}_\Psi^k (\mathbf{w} - \mathbf{w}^k),
\end{aligned} \tag{37}$$

with \mathbf{H}_Φ^k and \mathbf{H}_Ψ^k being the PSD approximating matrixes for $-\nabla^2\phi_3(\mathbf{w}^k)$ and $\nabla^2\phi_4(\mathbf{w}^k)$. $\eta(\mathbf{w}^k, \delta^k)$ can be computed from $\eta(\mathbf{w}^k, \delta^k)$

$$\begin{aligned}
&\triangleq (1 - \theta) \max_{j=3,4} \left\{ g_j(\mathbf{w}^k, \delta^k)_+ \right\} \\
&\quad + \theta \min_{\mathbf{w}, \delta} \left\{ \max_{j=3,4} \left\{ \tilde{g}_j(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k)_+ \right\} \mid (\mathbf{w}, \delta) \in \tilde{\mathcal{W}} \right\},
\end{aligned} \tag{38}$$

where $\tilde{\mathcal{W}}$ is a convex set defined as

$$\tilde{\mathcal{W}} = \{(\mathbf{w}, \delta) \mid \mathbf{w} \in \mathcal{W}, \delta \geq 0, g_i(\mathbf{w}, \delta) \leq 0, i = 1, 2, 5\}. \tag{39}$$

The second term in equation (38) is obtained as t from solving the following problem:

$$\begin{aligned}
&\underset{\mathbf{w}, \delta, t}{\text{minimize}} \quad t \\
&\text{subject to} \quad \tilde{g}_3(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \leq t, \\
&\quad \tilde{g}_4(\mathbf{w}, \delta; \mathbf{w}^k, \delta^k) \leq t, \\
&\quad (\mathbf{w}, \delta) \in \tilde{\mathcal{W}}, t \geq 0.
\end{aligned} \tag{40}$$

Problems (36) and (40) are both convex QCQP problems and can be efficiently solved by the corresponding solvers. We call it the Q-MVSKT algorithm and give the complete description in Algorithm 5.

VII. COMPLEXITY AND CONVERGENCE ANALYSIS

A. Complexity Analysis

First of all, it should be noted that the memory complexity for solving the high-order portfolio optimization problem is $\mathcal{O}(N^4)$ as the kurtosis matrix Ψ is of dimension $N \times N^3$. For example, when $N = 200$, storing a complete Ψ takes almost 12 GB of memory. Thus it is impractical to solve a very large-scale high-order portfolio optimization problem due to the memory restriction. All the algorithms investigated or proposed in this paper are iterative methods. Therefore, we discuss the computational complexity of constructing the surrogate problems in each

iteration, while the computational complexity of solving them depends on the specific solvers.

1) On Solving the MVSK Portfolio Problem (10): For Algorithms 1 and 2, the per-iteration computational cost of constructing the surrogate problems comes mainly from computing the gradients, which is $\mathcal{O}(N^4)$. For Algorithm 3, it is mainly from computing the gradient $\nabla f_{\text{ncvx}}(\mathbf{w}^k)$ and Hessian $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k)$, which in principle are $\mathcal{O}(N^4)$ and $\mathcal{O}(N^5)$, respectively. However, we can simplify the computation by first computing $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k) = -\lambda_3 \nabla^2 \phi_3(\mathbf{w}^k) + \lambda_4 \nabla^2 \phi_4(\mathbf{w}^k)$ as

$$\nabla^2 \phi_3(\mathbf{w}) = 6\Phi(\mathbf{I} \otimes \mathbf{w}) = 6 \left[\Phi^{(1)} \mathbf{w} \dots \Phi^{(N)} \mathbf{w} \right], \tag{41}$$

$$\begin{aligned}
\nabla^2 \phi_4(\mathbf{w}) &= 12\Phi(\mathbf{I} \otimes \mathbf{w} \otimes \mathbf{w}) \\
&= 12 \left[\Psi^{(1)}(\mathbf{w} \otimes \mathbf{w}) \dots \Psi^{(N)}(\mathbf{w} \otimes \mathbf{w}) \right],
\end{aligned} \tag{42}$$

where $\Phi^{(i)}$ is the i -th block matrix of dimension $N \times N$ in Φ and $\Psi^{(i)}$ is the i -th block matrix of dimension $N \times N^2$ in Ψ . Then the computational complexity of computing $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k)$ is reduced to $\mathcal{O}(N^4)$. With the usage of Corollary 2, $\nabla f_{\text{ncvx}}(\mathbf{w}^k)$ can be easily computed as

$$\nabla f_{\text{ncvx}}(\mathbf{w}^k) = -\frac{\lambda_3}{2} \nabla^2 \phi_3(\mathbf{w}^k) \mathbf{w}^k + \frac{\lambda_4}{3} \nabla^2 \phi_4(\mathbf{w}^k) \mathbf{w}^k. \tag{43}$$

Then the overall computational complexity of $\nabla f_{\text{ncvx}}(\mathbf{w}^k)$ and $\nabla^2 f_{\text{ncvx}}(\mathbf{w}^k)$ is still $\mathcal{O}(N^4)$. Therefore, the per-iteration computational cost of constructing the surrogate problems for Algorithms 1, 2, and 3 are $\mathcal{O}(N^4)$.

2) On Solving the MVSK Tilting Portfolio Problem (11): The per-iteration computational cost of constructing the surrogate problems in Algorithm 4 comes mainly from computing the gradients, while that in Algorithm 5 from computing both the gradients and Hessian. Similar to the above analysis, the latter can be simplified so that both algorithms admit the $\mathcal{O}(N^4)$ complexity on constructing the surrogate problems at each iteration.

B. Convergence Analysis

The convergence properties for the proposed algorithms are given in the following.

Proposition 6: Every limit point of the solution sequence $\{\mathbf{w}^k\}$ generated by the Algorithm 2 is a stationary point of problem (10).

Proof: Note that: 1) $\check{f}(\mathbf{w}, \mathbf{w}^k)$ is continuous in both \mathbf{w} and \mathbf{w}^k ; 2) $\check{f}(\mathbf{w}, \mathbf{w}^k)$ is a global upper bound function for $f(\mathbf{w})$ and is tangent to it at \mathbf{w}^k . Thus, [23, Assumption 1] is satisfied, and the proof of Proposition 6 follows directly from [23, Theorem 1]. ■

Proposition 7: Suppose $\gamma^k \in (0, 1]$, $\gamma^k \rightarrow 0$ and $\sum_k \gamma^k = +\infty$, and let $\{\mathbf{w}^k\}$ be the sequence generated by Algorithm 3. Then either Algorithm 3 converges in a finite number of iterations to a stationary point of (10) or every limit of $\{\mathbf{w}^k\}$ (at least one such point exists) is a stationary point of (10).

Proof: Note that the surrogate problem in Algorithm 3 only approximates the objective of problem (10) with a quadratic one but leave the constraints untouched, and: 1) \mathcal{W} is a compact and convex set; 2) $f(\mathbf{w})$ is continuously differentiable and coercive

on \mathcal{W} ; 3) $\nabla f_{\mathbf{w}}$ is Lipschitz continuous on \mathcal{W} (provided by Lemma 4). Thus, [20, Assumptions A1-A4] are satisfied, and the proof of Proposition 7 follows directly from [20, Theorem 3]. ■

Proposition 8: Suppose $\gamma^k \in (0, 1]$, $\gamma^k \rightarrow 0$ and $\sum_k \gamma^k = +\infty$, and let $\{\mathbf{w}^k\}$ be the sequence generated by Algorithm 4 or Algorithm 5. Then $\{\mathbf{w}^k\}$ is a generalized stationary point of the problem (27).

Proof: The only difference between Algorithm 4 and Algorithm 5 is that Algorithm 5 constructs the quadratic approximation for the non-convex constraints while Algorithm 4 simply linearizes all the constraints. However, it does not affect the convergence checking as they are both convex approximation for the constraints. Besides, it is easy to check that all the conditions in Assumption 1 are satisfied in both algorithms. Then the proof of Proposition 8 follows directly from [19]. ■

VIII. SOLVING OTHER HIGH-ORDER PORTFOLIO PROBLEMS

The algorithm framework proposed in this paper can be easily employed to solve other high-order portfolio problems.

A. MVSK Tilting Portfolio With General Deterioration Measures

As in [12], the MVSK tilting portfolio problem with general deterioration constraint is given as follows:

$$\begin{aligned} & \underset{\mathbf{w}, \delta}{\text{minimize}} \quad -\delta \\ & \text{subject to} \quad g_{\text{det}}(\mathbf{w}) \leq \kappa, \\ & \quad g_i(\mathbf{w}, \delta) \leq 0, \quad i = 1, \dots, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \delta \geq 0, \end{aligned} \quad (44)$$

where $g_{\text{det}}(\mathbf{w})$ is an assigned deterioration measure function and κ determines the maximum deterioration. As an example, $g_{\text{det}}(\mathbf{w})$ may be chosen as the deterioration of current \mathbf{w} from an ideal risk parity portfolio, i.e., the risk concentration [27]:

$$g_{\text{det}}(\mathbf{w}) = \sum_{i=1}^N \left(\frac{w_i(\Sigma \mathbf{w})_i}{\mathbf{w}^T \Sigma \mathbf{w}} - \frac{1}{N} \right)^2. \quad (45)$$

The regularized MVSK tilting portfolio problem is obtained by transforming the general deterioration constraint of problem (44) to a regularization term in the objective:

$$\begin{aligned} & \underset{\mathbf{w}, \delta}{\text{minimize}} \quad -\delta + \lambda g_{\text{det}}(\mathbf{w}) \\ & \text{subject to} \quad g_i(\mathbf{w}, \delta) \leq 0, \quad i = 1, \dots, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \delta \geq 0. \end{aligned} \quad (46)$$

Obviously, problems (44) and (46) are both solvable via the proposed algorithm framework in Section VI. The only difference is that here we also need to construct the convex approximating function for $g_{\text{ref}}(\mathbf{w})$ if it is non-convex. The procedure is trivial and hence omitted.

B. General Minkovski Distance MVST Portfolio

The general Minkovski distance MVST portfolio [28] admits the formulation

$$\begin{aligned} & \underset{\mathbf{w}, \mathbf{d}}{\text{minimize}} \quad z(\mathbf{d}) = \left(\sum_{k=1}^4 \left| \frac{d_k}{z_k} \right|^p \right)^{1/p} \\ & \text{subject to} \quad y_i(\mathbf{w}, \mathbf{d}) \leq 0, \quad i = 1, \dots, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \mathbf{d} \geq \mathbf{0}, \end{aligned} \quad (47)$$

where z_k is the aspired levels for k -th moments and

$$\begin{aligned} y_1(\mathbf{w}, \mathbf{d}) &= -\phi_1(\mathbf{w}) - d_1 + z_1, \\ y_2(\mathbf{w}, \mathbf{d}) &= \phi_2(\mathbf{w}) - d_2 - z_2, \\ y_3(\mathbf{w}, \mathbf{d}) &= -\phi_3(\mathbf{w}) - d_3 + z_3, \\ y_4(\mathbf{w}, \mathbf{d}) &= \phi_4(\mathbf{w}) - d_4 - z_4. \end{aligned} \quad (48)$$

It is easy to write a sequence of convex approximating surrogate problem as

$$\begin{aligned} & \underset{\mathbf{w}, \mathbf{d}}{\text{minimize}} \quad \nabla z(\mathbf{d}^k)^T (\mathbf{d} - \mathbf{d}^k) + \frac{\tau_{\mathbf{d}}}{2} \|\mathbf{d} - \mathbf{d}^k\|_2^2 \\ & \quad + \frac{\tau_{\mathbf{w}}}{2} \|\mathbf{w} - \mathbf{w}^k\|_2^2 \\ & \text{subject to} \quad \tilde{y}_i(\mathbf{w}, \mathbf{d}; \mathbf{w}^k, \mathbf{d}^k) \leq \eta(\mathbf{w}^k, \mathbf{d}^k), \quad i = 1, \dots, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \mathbf{d} \geq \mathbf{0}, \end{aligned}$$

where $\tilde{y}_i(\mathbf{w}, \mathbf{d}; \mathbf{w}^k, \mathbf{d}^k)$ is the convex approximation of $y_i(\mathbf{w}, \mathbf{d})$ at $(\mathbf{w}^k, \mathbf{d}^k)$, which can be easily constructed following similar procedures as in Section VI.

C. Polynomial Goal Programming MVST Portfolio

The polynomial goal programming (PGP) model for solving the high-order portfolio [29], [30] is a variation of the general Minkovski distance MVST portfolio taking investors' relative preference into consideration. It is formulated as

$$\begin{aligned} & \underset{\mathbf{w}, \mathbf{d}}{\text{minimize}} \quad z(\mathbf{d}) = \left| \frac{d_1}{z_1} \right|^{\lambda_1} + \left| \frac{d_2}{z_2} \right|^{\lambda_2} + \left| \frac{d_3}{z_3} \right|^{\lambda_3} + \left| \frac{d_4}{z_4} \right|^{\lambda_4} \\ & \text{subject to} \quad y_i(\mathbf{w}, \mathbf{d}) \leq 0, \quad i = 1, \dots, 4, \\ & \quad \mathbf{w} \in \mathcal{W}, \mathbf{d} \geq \mathbf{0}. \end{aligned} \quad (49)$$

This problem can still be easily handled via a similar procedure to solving the general Minkovski distance MVST portfolio.

IX. NUMERICAL EXPERIMENTS

In this section, we perform the numerical experiments on our proposed algorithms.¹ The data is obtained according to the following steps:

- 1) randomly select N stocks from a dataset of 500 stocks, each of them listed in the S&P 500 Index components;

¹We have released an R package `highOrderPortfolios` implementing our proposed algorithms at <https://github.com/dppalomar/highOrderPortfolios>.

- 2) randomly pick 5 N continuous trading days from 2004-01-01 to 2018-12-31;
- 3) compute four sample moments of the selected N stocks during the picked trading period.

The starting point is selected as $\mathbf{w}^0 = \frac{1}{N}\mathbf{1}$ for all methods. Without loss of generality, we simply set $L = 1$ (no leverage), $\theta = \frac{1}{2}$, and choose the diminishing step size sequence as:

$$\gamma^0 = 1, \quad \gamma^k = \gamma^{k-1} (1 - 10^{-2}\gamma^k). \quad (50)$$

The inner solvers for QP, LP, and QCQP are selected as `quadprog` [31], `lpSolveAPI` [32], and `ECOS` [33], [34], respectively. The algorithm is regarded as converged when any of the following conditions is satisfied:

$$\begin{aligned} |\mathbf{x}^{k+1} - \mathbf{x}^k| &\leq 10^{-6} (|\mathbf{x}^{k+1}| + |\mathbf{x}^k|), \\ |f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| &\leq 10^{-6} (|f(\mathbf{x}^{k+1})| + |f(\mathbf{x}^k)|). \end{aligned} \quad (51)$$

A. On the MVSK Portfolio Problem (10)

We first set $N = 100$ and then solve the problem (10) using benchmark DC-based Algorithm 1, our proposed MM-based Algorithm 2, and our proposed Q-MVSK Algorithm 3. The weights for the four moments are decided according to the fourth order expansion of the Constant Relative Risk Aversion (CRRA) utility function:

$$\begin{aligned} \lambda_1 &= 1, & \lambda_2 &= \frac{\xi}{2}, \\ \lambda_3 &= \frac{\xi(\xi+1)}{6}, & \lambda_4 &= \frac{\xi(\xi+1)(\xi+2)}{24}, \end{aligned} \quad (52)$$

where $\xi \geq 0$ is the risk aversion parameter [9] and set to be 10 in our experiments. For comparison, we also solve the problem via the following benchmark methods:

- 1) differential evolution (DE): similar to [9], we can solve the high-order portfolio using differential evolution via R package `DEoptim` [35];
- 2) genetic algorithm (GA): similar to [10], we can solve the high-order portfolio using genetic algorithms via R package `GA` [36];
- 3) NLOpt: a optimization tool box for nonlinear optimization. We use it via the R package `nloptr` [37] with gradients passed.

The DC-SOS method is excluded here since its initialization and solution take much more time than all methods.² We use the same randomly generated starting population of size 100 for DE and GA methods. Besides, we allow the DE and GA methods to run in parallel, i.e., making full use of all 6 cores of our laptop. While other methods are run on a single core. In Fig. 4, we compare the convergence of these algorithms. Significantly, the Q-MVSK algorithm can converge to the best result in very few iterations, which is much more efficient than the solver `nloptr`. Even with parallel mode, the two metaheuristic optimization tools, e.g., DE and GA, do not perform well in this problem.

²We acknowledge the author of [16] for sharing their code, which are publicly accessible at <https://github.com/niuyishuai/MVSKOPT>, <https://github.com/niuyishuai/DCAM>, and <https://github.com/niuyishuai/Polylab>.

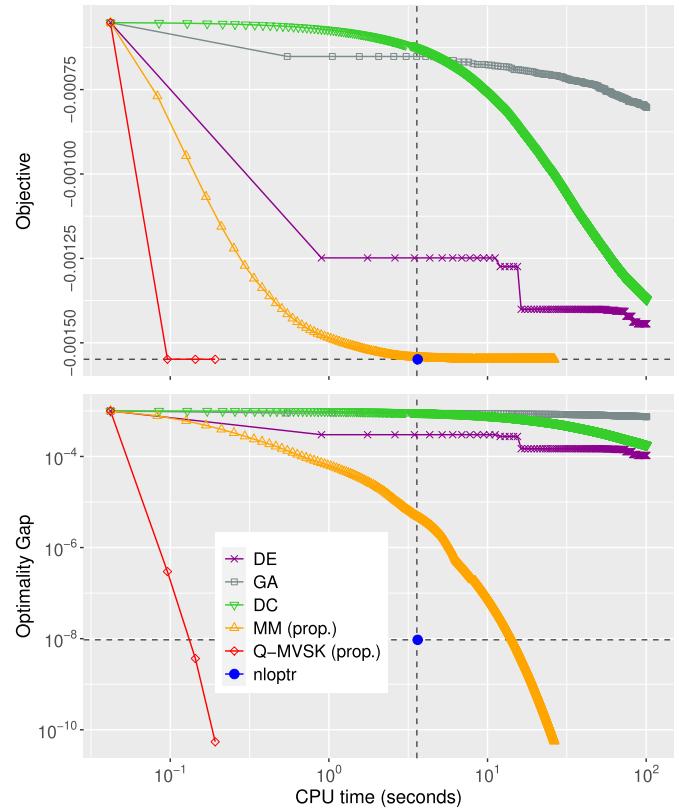


Fig. 4. Convergence of algorithms on solving MVSK problem (10) with $N = 100$.

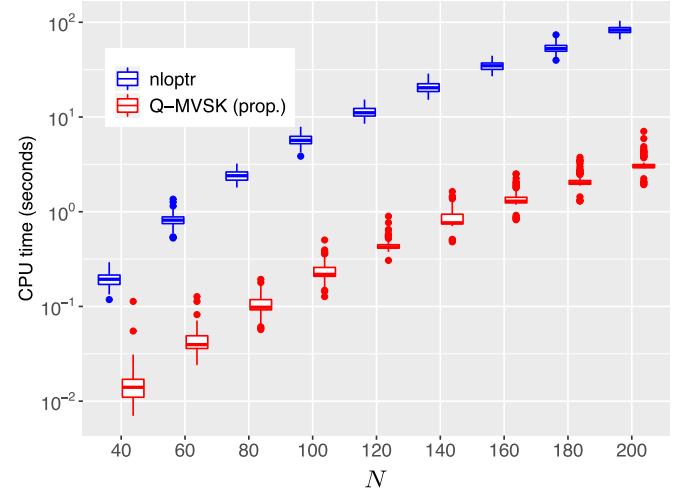


Fig. 5. Time usage of algorithms on solving MVSK problem (10).

It can be easily explained as both of them involve frequently computing the objective value, which is very expensive in this case. The DC-based and MM-based algorithms are both slower than the general solver `nloptr`. It implies that they may use very loose upper bounds. The MM-based algorithm, though much faster than the DC-based algorithm, is far from being comparable with the Q-MVSK algorithm.

In Fig. 5, we show the comparison of time consumption of the proposed Q-MVSK algorithm and `nloptr` while changing the problem dimension N . The DC-based and the MM-based

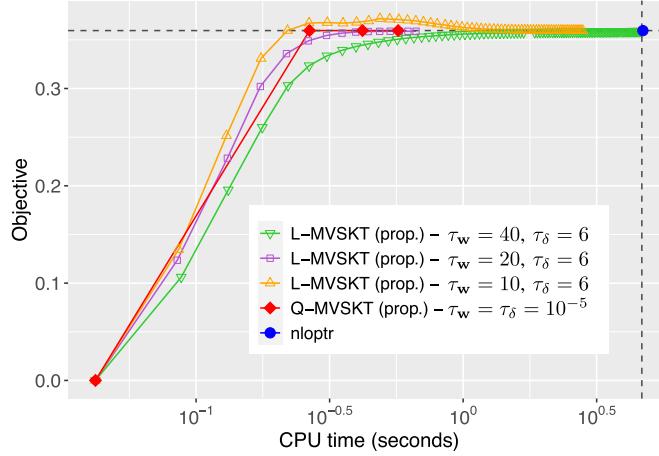


Fig. 6. Convergence of proposed algorithms for MVSK tilting problem (11) with $N = 100$ and $\kappa = 0.3\sqrt{\phi_2(\mathbf{w}_0)}$.

algorithms are not included as they are too slow to be compared with the proposed Q-MVSK algorithm and `nloptr`. For fair comparison, we force `nloptr` to run until it reaches the objective obtained from Q-MVSK algorithm. The result is obtained by performing the experiments on 100 realizations of randomly generated data. We can see that our proposed Q-MVSK algorithm is consistently more than one order of magnitude faster than `nloptr`.

B. On the MVSK Tilting Portfolio Problem (11)

Similar to the above, we first set $N = 100$ and then solve the problem (11) via the proposed Algorithms 4 and 5, respectively. The reference portfolio is simply chosen as the equally weighted portfolio, i.e.,

$$\mathbf{w}_0 = \frac{1}{N} \mathbf{1}. \quad (53)$$

The tilting direction \mathbf{d} is decided as $d_i = |\phi_i(\mathbf{w}_0)|$. We choose κ in (11) as $\kappa = c \times \sqrt{\phi_2(\mathbf{w}_0)}$ with $c \geq 0$. The general solver `nloptr` is also included for comparison.³ We find that, although the final convergence is guaranteed, the fast convergence of the proposed L-MVSKT algorithm really relies on the proper choice of τ_w and τ_δ , while that of our proposed Q-MVSKT is much robust. For example, in Fig. 6, we set $\kappa = 0.3\sqrt{\phi_2(\mathbf{w}_0)}$ and show the convergence of the proposed algorithms. It is significant that the Q-MVSKT algorithm converges in few iterations simply with $\tau_w = \tau_\delta = 10^{-5}$. The L-MVSKT algorithm can also converge with comparable speed when parameters are properly tuned. It may be explained as that the L-MVSKT algorithm poorly approximates all constraints by linear functions, making the solution to approximating problems easily violate the original constraints. However, the Q-MVSKT algorithm preserves the convex constraints and approximates the non-convex constraints by convex quadratic functions, which turns out to work very well. Besides, we notice that solving the QCQP problem is significantly slower than solving the QP problem of the same size. In Fig. 7, we show the final results of these algorithms when

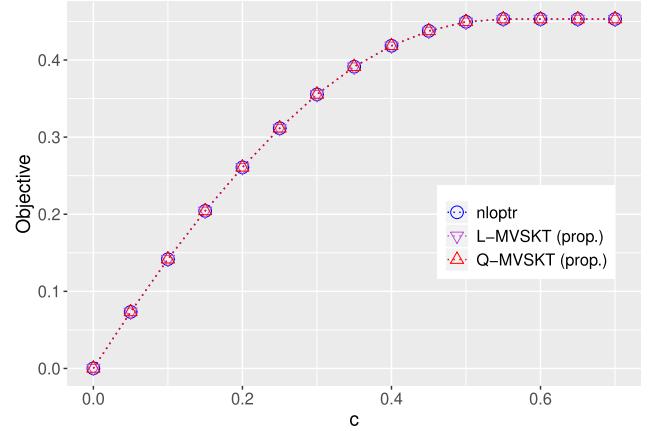


Fig. 7. Comparison of the results with different κ ($\kappa = c \times \sqrt{\phi_2(\mathbf{w}_0)}$).

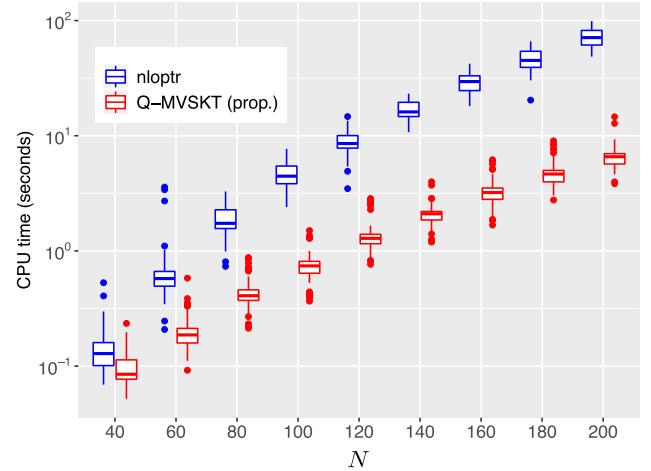


Fig. 8. Time usage of algorithms on solving problem (11).

changing the maximum tracking error constraint. It is clear that all algorithms can give the same results, which are nondecreasing when κ increases. In Fig. 8, we show the comparison of time consumption of the proposed Q-MVSKT algorithm and `nloptr` while changing the problem dimension N . The proposed L-MVSKT algorithm is not included as its convergence speed relies heavily on parameter tuning. The result is obtained by performing the experiments on 100 realizations of randomly generated data. It is significant that our proposed Q-MVSKT algorithm is consistently around one order of magnitude faster than `nloptr`.

X. CONCLUSION

In this paper, we have considered the high-order moments of the portfolio return for high-order portfolio optimization. We have proposed an efficient algorithm framework for solving high-order portfolio optimization problems based on the successive convex approximation framework. In particular, we have proposed efficient algorithms for solving the mean-variance-skewness-kurtosis portfolio optimization problem and the mean-variance-skewness-kurtosis tilting portfolio optimization problem. Theoretically, all the proposed algorithms enjoy convergence to a stationary point. Extensive numerical experiments show that our proposed algorithms, specifically the Q-MVSK

³We use directly the implementation from authors of [12], which is available at <https://github.com/cdries/mvskPortfolios>.

and Q-MVSKT algorithms, are much more efficient than the existing methods and simply using a general solver.

APPENDIX

A. Proof for Lemma 1

According to the Leibniz integral rule [38], we have

$$\begin{aligned}\nabla \phi_3(\mathbf{w}) &= \frac{\partial \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w}]}{\partial \mathbf{w}} \\ &= \mathbb{E}\left[\frac{\partial(\mathbf{w}^T \tilde{\mathbf{r}} \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w})}{\partial \mathbf{w}}\right] \\ &= \mathbb{E}[3\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{w} \otimes \mathbf{w})] \\ &= 3\mathbb{E}[\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{w} \otimes \mathbf{w})] \\ &= 3\Phi(\mathbf{w} \otimes \mathbf{w}),\end{aligned}\quad (54)$$

$$\begin{aligned}\nabla \phi_4(\mathbf{w}) &= \frac{\partial \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w}]}{\partial \mathbf{w}} \\ &= \mathbb{E}\left[\frac{\partial(\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w})}{\partial \mathbf{w}}\right] \\ &= \mathbb{E}[4\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w})] \\ &= 4\mathbb{E}[\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w})] \\ &= 4\Psi(\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w}),\end{aligned}\quad (55)$$

$$\begin{aligned}\nabla^2 \phi_3(\mathbf{w}) &= \frac{\partial^2 \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w}]}{\partial \mathbf{w} \partial \mathbf{w}^T} \\ &= \mathbb{E}\left[\frac{\partial^2(\mathbf{w}^T \tilde{\mathbf{r}} \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T}\right] \\ &= \mathbb{E}[6\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{I} \otimes \mathbf{w})] \\ &= 6\mathbb{E}[\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{I} \otimes \mathbf{w})] \\ &= 6\Phi(\mathbf{I} \otimes \mathbf{w}),\end{aligned}\quad (56)$$

$$\begin{aligned}\nabla^2 \phi_4(\mathbf{w}) &= \frac{\partial^2 \mathbb{E}[\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w}]}{\partial \mathbf{w} \partial \mathbf{w}^T} \\ &= \mathbb{E}\left[\frac{\partial^2(\mathbf{w}^T \tilde{\mathbf{r}} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w} \tilde{\mathbf{r}}^T \mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T}\right] \\ &= \mathbb{E}[12\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{I} \otimes \mathbf{w} \otimes \mathbf{w})] \\ &= 12\mathbb{E}[\tilde{\mathbf{r}}(\tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T \otimes \tilde{\mathbf{r}}^T)(\mathbf{I} \otimes \mathbf{w} \otimes \mathbf{w})] \\ &= 12\Psi(\mathbf{I} \otimes \mathbf{w} \otimes \mathbf{w}).\end{aligned}\quad (57)$$

B. Proof for Lemma 4

According to the Gershgorin circle theorem [39], we have

$$\begin{aligned}\rho(\nabla^2 f_{\text{ncvx}}(\mathbf{w})) &\leq \|\nabla^2 f_{\text{ncvx}}(\mathbf{w})\|_\infty \\ &\leq \lambda_3 \|\nabla^2 \phi_3(\mathbf{w})\|_\infty + \lambda_4 \|\nabla^2 \phi_4(\mathbf{w})\|_\infty.\end{aligned}\quad (58)$$

Under the constraints in (9), we can get

$$\begin{aligned}\|\nabla^2 \phi_3(\mathbf{w})\|_\infty &= 6 \max_{1 \leq i \leq N} \sum_{j=1}^N \left| \sum_{k=1}^N \Phi_{ij}^{(k)} w_k \right| \\ &\leq 6 \max_{1 \leq i \leq N} \sum_{j=1}^N \sum_{k=1}^N |\Phi_{ij}^{(k)}| |w_k| \\ &\leq 6 \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k \leq N} L |\Phi_{ij}^{(k)}| \\ &= 6L \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k \leq N} |\Phi_{ij}^{(k)}|,\end{aligned}\quad (59)$$

$$\begin{aligned}\|\nabla^2 \phi_4(\mathbf{w})\|_\infty &= 12 \max_{1 \leq i \leq N} \sum_{j=1}^N \left| \sum_{k=1}^N w_k \sum_{l=1}^N \Psi_{ij}^{(k,l)} w_l \right| \\ &\leq 12 \max_{1 \leq i \leq N} \sum_{j=1}^N \sum_{k=1}^N |w_k| \sum_{l=1}^N |\Psi_{ij}^{(k,l)}| |w_l| \\ &\leq 12 \max_{1 \leq i \leq N} \sum_{j=1}^N \sum_{k=1}^N |w_k| \max_{1 \leq l \leq N} L |\Psi_{ij}^{(k,l)}|, \\ &\leq 12 \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k \leq N} L \max_{1 \leq l \leq N} L |\Psi_{ij}^{(k,l)}| \\ &= 12L^2 \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k, l \leq N} |\Psi_{ij}^{(k,l)}|.\end{aligned}\quad (60)$$

Therefore, we have

$$\begin{aligned}\rho(\nabla^2 f_{\text{ncvx}}(\mathbf{w})) &\leq 6\lambda_3 L \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k \leq N} |\Phi_{ij}^{(k)}| \\ &\quad + 12\lambda_4 L^2 \max_{1 \leq i \leq N} \sum_{j=1}^N \max_{1 \leq k, l \leq N} |\Psi_{ij}^{(k,l)}|.\end{aligned}\quad (61)$$

REFERENCES

- [1] H. Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [2] P. N. Kolm, R. Tütüncü, and F. J. Fabozzi, "60 years of portfolio optimization: Practical challenges and current trends," *Eur. J. Oper. Res.*, vol. 234, no. 2, pp. 356–371, 2014.
- [3] C. Adcock, M. Eling, and N. Loperfido, "Skewed distributions in finance and actuarial science: A review," *The Eur. J. Finance*, vol. 21, no. 13–14, pp. 1253–1281, 2015.
- [4] S. I. Resnick, *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Berlin, Germany: Springer Science & Business Media, 2007.
- [5] C. R. Harvey and A. Siddique, "Conditional skewness in asset pricing tests," *The J. Finance*, vol. 55, no. 3, pp. 1263–1295, 2000.
- [6] N. J. Jobst and S. A. Zenios, "The tail that wags the dog: Integrating credit risk in asset portfolios," *J. Risk Finance*, pp. 31–43, 2001.
- [7] A. Ang, J. Chen, and Y. Xing, "Downside risk," *The Rev. Financial Stud.*, vol. 19, no. 4, pp. 1191–1239, 2006.

[8] T. P. Dinh and Y.-S. Niu, "An efficient DC programming approach for portfolio decision with higher moments," *Comput. Optim. Appl.*, vol. 50, no. 3, pp. 525–554, 2011.

[9] K. Boudt, W. Lu, and B. Peeters, "Higher order moments of multifactor models and asset allocation," *Finance Res. Lett.*, vol. 13, pp. 225–233, 2015.

[10] S. Kshatriya and P. K. Prasanna, "Genetic algorithm-based portfolio optimization with higher moments in global stock markets," *J. Risk*, vol. 20, no. 4, pp. 1–26, 2018.

[11] W. H. Jean, "The extension of portfolio analysis to three or more parameters," *J. Financial Quantitative Anal.*, vol. 6, no. 1, pp. 505–515, 1971.

[12] K. Boudt, D. Cornilly, F. V. Holle, and J. Willems, "Algorithmic portfolio tilting to harvest higher moment gains," *Heliyon*, vol. 6, no. 3, 2020, Art. no. e03516.

[13] K. G. Murty and S. N. Kabadi, "Some NP-complete problems in quadratic and nonlinear programming," *Math. Program.*, vol. 39, no. 2, pp. 117–129, 1987.

[14] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[15] A. Savine, *Modern Computational Finance: AAD and Parallel Simulations*. Hoboken, NJ, USA: Wiley, 2018.

[16] Y.-S. Niu and Y.-J. Wang, "Higher-order moment portfolio optimization via the difference-of-convex programming and sums-of-squares," 2019, *arXiv:1906.01509*.

[17] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[18] G. Scutari and Y. Sun, "Parallel and distributed successive convex approximation methods for big-data optimization," in *Multi-Agent Optim.: Cetraro, Italy 2014*, F. Facchinei and J.-S. Pang, Eds. Berlin, Germany: Springer, 2018, ch. 3, pp. 141–308.

[19] F. Facchinei, V. Kungurtsev, L. Lampariello, and G. Scutari, "Ghost penalties in nonconvex constrained optimization: Diminishing stepsizes and iteration complexity," *Mathematics Operations Research to appear*, 2020.

[20] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[21] Z. Zhao, R. Zhou, and D. P. Palomar, "Optimal mean-reverting portfolio with leverage constraint for statistical arbitrage in finance," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1681–1695, Apr. 2019.

[22] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization—Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.

[23] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

[24] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The Amer. Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[25] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear Algebra Appl.*, vol. 103, pp. 103–118, 1988.

[26] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[27] Y. Feng and D. P. Palomar, "SCRIP: Successive convex optimization methods for risk parity portfolio design," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5285–5300, Oct. 2015.

[28] P. Nijkamp and J. Spronk, "Interactive multiple goal programming: An evaluation and some results," in *Multiple Criteria Decision Making Theory and Applications*. Berlin, Germany: Springer, 1980, pp. 278–293.

[29] K. K. Lai, L. Yu, and S. Wang, "Mean-variance-skewness-kurtosis-based portfolio optimization," in *Proc. 1st Int. Multi-Symp. Comput. Comput. Sci.*, vol. 2, 2006, pp. 292–297.

[30] M. Aksarayli and O. Pala, "A polynomial goal programming model for portfolio optimization based on entropy and higher moments," *Expert Syst. Appl.*, vol. 94, pp. 185–192, 2018.

[31] B. A. Turlach and A. Weingessel, *Quadprog: Functions to Solve Quadratic Programming Problems*, 2019, R package version 1.5-7. Accessed: Jul. 8, 2020. [Online]. Available: <https://CRAN.R-project.org/package=quadprog>

[32] K. Konis and F. Schwendinger, *lpSolveAPI: R. Interface to 'lp_solve' Version 5.5.2.0*, 2020, R package version 5.5.2.0-17.6. Accessed: Jul. 8, 2020. [Online]. Available: <https://CRAN.R-project.org/package=lpSolveAPI>

[33] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Proc. Eur. Control Conf.*, 2013, pp. 3071–3076.

[34] A. Fu and B. Narasimhan, *ECOSolveR: Embedded Conic Solver in R*, 2019, R package version 0.5.3. Accessed: Jul. 8, 2020. [Online]. Available: <https://CRAN.R-project.org/package=ECOSolveR>

[35] D. Ardia, K. Boudt, P. Carl, K. Mullen, and B. G. Peterson, "Differential evolution with DEoptim: An application to non-convex portfolio optimization," *The R. J.*, vol. 3, no. 1, pp. 27–34, 2011.

[36] L. Scrucca, "GA: A package for genetic algorithms in R," *J. Stat. Softw.*, vol. 53, no. 4, pp. 1–37, 2013.

[37] J. Ypma and S. G. Johnson, *Introduction to Nloptr: An R. Interface to Nlopt*, 2020, R package version 1.2.2.1. Accessed: Jul. 8, 2020. [Online]. Available: <https://CRAN.R-project.org/package=nloptr>

[38] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Araphs, and Mathematical Tables*. Dover Publications, 1965.

[39] R. S. Varga, *Geršgorin and His Circles*. Berlin, Germany: Springer Science & Business Media, 2010.



Rui Zhou (Student Member, IEEE) received the B.Eng. degree in information engineering from Southeast University, Nanjing, China, in 2017. He is currently working toward the Ph.D. degree with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong.

His research interests include optimization algorithms, statistical signal processing, machine learning, and financial engineering.



Daniel P. Palomar (Fellow, IEEE) received the Electrical Engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively. From 2004 to 2006, he was a Fulbright Scholar with Princeton University, Princeton, NJ, USA.

He is currently a Professor with the Department of Electronic and Computer Engineering and the Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology (HKUST), Hong Kong, which he joined in 2006. He had previously held several research appointments, including King's College London, London, U.K., Stanford University, Stanford, CA, USA, Telecommunications Technological Center of Catalonia, Barcelona, Spain, Royal Institute of Technology, Stockholm, Sweden, University of Rome La Sapienza, Rome, Italy, and Princeton University.

His current research interests include applications of optimization theory, graph methods, and signal processing in financial systems and big data analytics.

Dr. Palomar was the recipient of the 2004/06 Fulbright Research Fellowship, the 2004, 2015, and 2020 (coauthor) Young Author Best Paper Awards by the IEEE Signal Processing Society, the 2015–16 HKUST Excellence Research Award, the 2002/03 Best Ph.D. prize in Information Technologies and Communications by the Technical University of Catalonia, the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in information technologies and communications by the Epson Foundation, and the 2004 prize for the Best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT.

He has been the Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING 2016 Special Issue on Financial Signal Processing and Machine Learning for Electronic Trading, the IEEE Signal Processing Magazine 2010 Special Issue on Convex Optimization for Signal Processing, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2008 Special Issue on Game Theory in Communication Systems, and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2007 Special Issue on Optimization of MIMO Transceivers for Realistic Communication Networks, and an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.