AN OPTIMAL TRADE-OFF MODEL FOR PORTFOLIO SELECTION WITH SENSITIVITY OF PARAMETERS

YANQIN BAI*, YUDAN WEI AND QIAN LI

Department of Mathematics
Shanghai University
Shanghai 200444, China

(Communicated by Kok Lay Teo)

Abstract. In this paper, we propose an optimal trade-off model for portfolio selection with sensitivity of parameters, which are estimated from historical data. Mathematically, the model is a quadratic programming problem, whose objective function contains three terms. The first term is a measurement of risk. And the later two are the maximum and minimum sensitivity, which are non-convex and non-smooth functions and lead to the whole model to be an intractable problem. Then we transform this quadratic programming problem into an unconstrained composite problem equivalently. Furthermore, we develop a modified accelerated gradient (AG) algorithm to solve the unconstrained composite problem. The convergence and the convergence rate of our algorithm are derived. Finally, we perform both the empirical analysis and the numerical experiments. The empirical analysis indicates that the optimal trade-off model results in a stable return with lower risk under the stress test. The numerical experiments demonstrate that the modified AG algorithm outperforms the existed AG algorithm for both CPU time and the iterations, respectively.

1. Introduction. In 1952 Markowitz [17] proposed the mean-variance model for portfolio selection and opened a new era of quantitative analysis in the portfolio selection. The mean-variance model seeks to reduce the total variance of the portfolio with the certain level of the expected return. For mean-variance model, the return is measured by the mean value and the risk is quantified by the variance of the portfolio return, respectively. Based on Markowitz’s work, different extensions of the mean-variance model have been proposed, such as considering the maximum individual risk [25], marginal risk [27], probabilistic risk [24], cardinality constraint [23, 26] and so on [5].

Since the mean and the variance in the mean-variance model are measured with lacking and missing historical data, it may cause the uncertainty of estimation errors. The accuracy of the parameters in the mean-variance model directly affects both measurements of return and risk. Chopra et al. [4] investigated the effectiveness of estimation error of the mean, variance and covariance and indicated that...
even a small error of parameters might cause a large deviation for portfolio selection. To overcome the uncertainty of parameters, the popular and useful method is robust optimization, whose main idea is to consider the worst-case portfolio given an uncertainty interval of parameters. The seminal work was done by Goldfarb et al. [9] who proposed a robust mean-variance model for portfolio selection and reformulated it into a second-order cone programming problem. However, Scherer [22] pointed out that the robust mean-variance model was over conservation and could not bring the extra return. Oppositely, it might increase the computational cost. Recently, Cui [6] proposed a mean-variance model with sensitivity of parameters as a class of constraints for portfolio selection. In her model both upper and lower bound of parameters are given as constants to control the error of parameters. However, it is difficult to choose the upper and lower bounds for the parameters and it is still an optimization problem. Similarly, Li et al. [15] presented a trade-off mean-variance model between total risk and maximum relative marginal risk to avoid choosing parameters. Motivated by Cui and Li’s researches, we use the sensitivity of parameters defined in [6] to formulate a mean-variance model, in which we balance the risk and the sensitivity of parameters by a scalar to replace the choice of the upper and lower bounds of parameters.

Usually, the mean-variance models presented above are non-convex quadratically constrained quadratic programming (QCQP) problem. The methods to solve non-convex QCQP problems are mainly branch-and-bound algorithms. Basic references include Horst et al. [11], Al-Khayyal et al. [1], Raber [21], Audet et al. [2], and Linderoth [16]. Recently, Deng et al. [7] proposed a branch-and-cut algorithm to speed up the computational effort for a portfolio selection which is formulated into a semidefinite programming (SDP) problem. It is well-known that the branch-and-bound algorithm performances badly with large-scale optimization problems. Nesterov’s accelerated gradient (AG) algorithm [18] has attracted much attention recently partly due to the increasing demand solving large-scale convex programming (CP) problems by using the fast first-order algorithms. Then Nesterov [20] used AG algorithm and a new approach for constructing efficient schemes to solve a simple non-smooth CP program. Lan [14] further showed that the AG algorithm, when employed with proper stepsize policies, is optimal for solving not only smooth CP problems, but also general non-smooth and stochastic CP problems. More recently, Ghadimi et al. [8] developed a class of AG algorithms to solve the non-convex and stochastic optimal problems.

Inspired by the AG algorithm, in this paper, we propose an optimal trade-off model for portfolio selection with sensitivity of parameters, which are estimated from historical data. The feature of model is to balance the risk and the error caused by parameters. The sensitivity of parameters are measured by maximum and minimum sensitivity for all assets. Mathematically, the model is a quadratic programming problem, whose objective function contains three terms. The first term is a measurement of risk. And the later two are the maximum and minimum sensitivity, which are non-convex and non-smooth functions, and lead to the whole model to be an intractable problem. Then we transform this quadratic programming problem into an unconstrained composite problem equivalently. Furthermore, we develop a modified AG algorithm to solve the unconstrained composite problem. The convergence and the convergence rate of our algorithm are derived. Finally, we perform both the empirical analysis and the numerical experiments. The empirical analysis indicates that the optimal trade-off model results in a stable return with
lower risk under the stress test. The numerical experiments demonstrate that the modified AG algorithm outperforms the existed AG algorithm for both CPU time and the iterations, respectively.

The rest of this paper is organized as follows. In Section 2 we recall some preliminaries of this paper. In Section 3 we propose an optimal trade-off model for portfolio selection with sensitivity of parameters and transform it to an unconstrained composite problem. In Section 4 we develop a modified AG algorithm to solve the unconstrained composite problem. The empirical analysis is derived in Section 5. The numerical experiments are shown in Section 6. Finally, some concluding remarks are made in Section 7.

2. Preliminaries. In this section, we briefly recall some basic definitions and basic theorems that will be used below. We first recall the mean-variance model, the concept of the sensitivity of parameters and the portfolio selection model with the sensitivity of parameter constraints in [6]. Then we recall the definition of Lipschitz continuous, subgradient of a convex function in [19] and the AG algorithm in [8], respectively.

In the mean-variance model, consider $n$ risky assets with random rates of return $r = (r_1, r_2, \ldots, r_n)$ for an investor. Let $x = (x_1, x_2, \ldots, x_n)$ be the amount of the portfolio to be invested in the $n$ risky assets. Then the random return of portfolio $x$ is $R(x) = \langle r, x \rangle$, and the expected value and the variance are

$$
\mu(x) = E\langle r, x \rangle := \langle \mu, x \rangle, \quad \sigma^2(x) = E[\langle r, x \rangle - \mu(x)]^2 := x\Sigma x^T,
$$

where $\mu$ and $\Sigma$ are the mean vector and the covariance matrix of $r$. The Markowitz’s mean-variance model can be formulated as follows:

$$
\min \ x\Sigma x^T \quad \text{s.t.} \quad \langle \mu, x \rangle \geq \rho, \quad \langle e, x \rangle = 1, \quad x \geq 0, \quad (MV)
$$

where $\rho$ is the expected return of the investor and $e = (1, 1, \ldots, 1)$. The variance of portfolio selection can be rewritten as

$$
x\Sigma x^T = \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} \sigma_i \sigma_j x_i x_j,
$$

where $\rho_{ij}$ is the correlation coefficient among assets and $\sigma = (\sigma_1, \ldots, \sigma_n)$ is the vector of standard deviation. Cui [6] introduced the concept of sensitivity of parameters.

**Definition 2.1.** [6] The sensitivity of the parameters for asset $i$ is defined as the partial derivative of variance of portfolio $x$ with respect to the standard deviation of individual asset $\sigma_i$:

$$
S_i(x) := \frac{\partial x\Sigma x^T}{\partial \sigma_i} = 2 \sum_{j \neq i} \rho_{ij} \sigma_j x_i x_j + 2\sigma_i x_i^2 = x\Sigma_i x^T, \quad i = 1, \ldots, n,
$$

where

$$
\Sigma_i = \begin{pmatrix}
\rho_{i1} \sigma_1 \\
\vdots \\
\rho_{i1} \sigma_1 & \cdots & 2\sigma_i & \cdots & \rho_{in} \sigma_n \\
\vdots \\
\rho_{ni} \sigma_n
\end{pmatrix}.
$$
Moreover, as \( \frac{\partial \mu x^T}{\partial \mu} = x, \ i = 1, \ldots, n \), restricting the upper and lower bounds of \( x \) is also necessary. Given the estimated parameters \( \mu \) and \( \Sigma \), Cui \cite{Cui2010} developed a portfolio selection model with the sensitivity of parameters constraints, i.e.,

\[
\begin{align*}
\min & \ x \Sigma x^T \\
\text{s.t.} & \ \langle \mu, x \rangle \geq \rho, \ \langle e, x \rangle = 1, \ l \leq x \leq u, \\
& \ -k_i^2 \leq x_i x^T \leq s_i^2, \ i = 1, \ldots, m,
\end{align*}
\]

where \( m \leq n \) denotes the number of assets that the sensitivity of parameters should be restricted, \( k, s \) are the lower and upper bounds of sensitivity of parameters, and \( l, u \) are the lower and upper bounds of \( x \). Since \( \Sigma \) is an indefinite matrix, \( (\text{MV}_{sc}) \) is a non-convex quadratically constrained quadratic program.

**Definition 2.2.** \cite{Beck2010} Let \( f(x) \) be a differentiable function on \( \mathbb{R}^n \). The gradient function \( \nabla f \) is said to be Lipschitz continuous if there exists a constant \( L_f > 0 \) such that

\[
\| \nabla f(y) - \nabla f(x) \| \leq L_f \| y - x \|, \quad \forall x, y \in \mathbb{R}^n,
\]

where the norm \( \| \cdot \| \) denotes the Euclidean norm. We denote \( f(x) \in C^{1,1}_{L_f}(\mathbb{R}^n) \).

**Definition 2.3.** \cite{Beck2010} Let \( f(x) \) be a convex function. A vector \( g \) is called the subgradient of function \( f \) at point \( x_0 \in \text{dom } f \) if for any \( x \in \text{dom } f \), we have

\[
\psi(x) \geq \psi(x_0) + \langle g, x - x_0 \rangle.
\]

The subgradient set of function \( f \) at point \( x_0 \) is denoted as \( \partial f(x_0) \).

Consider a class of composite problems given by

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \ \Psi(x) + \chi(x), \\
\end{align*}
\]

where \( \Psi(x) := f(x) + h(x) \), \( f \in C^{1,1}_{L_f}(\mathbb{R}^n) \) is possibly nonconvex, \( h(x) \in C^{1,1}_{L_h}(\mathbb{R}^n) \) is convex, and \( \chi \) is a simple convex (possibly non-smooth) function with bounded domain.

**Lemma 2.4.** \cite{Ghadimi2020} If \( \Psi(\cdot) \) is defined in \( (CP) \), then for any \( x, y \in \mathbb{R}^n \), we have

\[
-L_f \frac{1}{2} \| y - x \|^2 \leq \Psi(y) - \Psi(x) - \langle \nabla \Psi(x), y - x \rangle \leq L_f \frac{1}{2} \| y - x \|^2.
\]

**Lemma 2.5.** \cite{Ghadimi2020} If \( \chi(\cdot) \) is a proper closed convex function with bounded domain, then there exists a constant \( M \) such that \( \| P(x, y, c) \| \leq M \) for any \( c \in (0, +\infty) \) and \( x, y \in \mathbb{R}^n \), where \( P(x, y, c) \) is given by

\[
\begin{align*}
P := & \ \arg \min_{w \in \mathbb{R}^n} \{ \langle y, w \rangle + \frac{1}{2c} \| w - x \|^2 + \chi(w) \}.
\end{align*}
\]

Ghadimi et al. \cite{Ghadimi2020} also introduced an important quantity that will be a termination criterion in the AG algorithm, i.e.,

\[
\mathcal{G} := \frac{1}{c} [x - \chi(x, y, c)]
\]
Algorithm 2.1. The AG algorithm [8]

**Step 0.** Input $x_0 \in \mathbb{R}^n$, $\{\alpha_k\}$ s.t. $\alpha_1 = 1$ and $\alpha_k \in \{0, 1\}$ for any $k \geq 2$, $\{\beta_k > 0\}$, $\{\lambda_k > 0\}$, and the accuracy parameter $\epsilon$. Let $x_0^{ag} = x_0$, $k = 1$.

**Step 1.** Let

$$x_{md_k} = (1 - \alpha_k)x_{ag_{k-1}} + \alpha_k x_{k-1}.$$  

**Step 2.** Compute $\nabla \Psi(x_{md_k})$, let

$$x_k = \mathcal{P}(x_{k-1}, \nabla \Psi(x_{md_k}), \lambda_k),$$

$$x_k^{ag} = \mathcal{P}(x_{md_k}, \nabla \Psi(x_{md_k}), \beta_k).$$

**Step 3.** If $\|G\| \leq \epsilon$, stop. Else $k = k + 1$, go to Step 1.

3. Optimal trade-off model and equivalent transformation. In this section, we first propose an optimal trade-off model for portfolio selection with sensitivity of parameters. Then we analyze the feature of model. Based on the properties of model, we finally transform it to an unconstrained composite problem equivalently.

3.1. Optimal trade-off model. In our model, we use the maximum and minimum sensitivity of parameters of all assets as a measure of sensitivity. The set of assets whose the sensitivity of parameters should be restricted is denoted as $S = \{1, \ldots, m\}$.

The objective function contains two parts in our model, the first one is the classical risk term denoted by $x \Sigma x^T$, and the second one is the sensitivity of parameters denoted by $\max_{i \in S} \{x \Sigma_i x^T\} - \min_{i \in S} \{x \Sigma_i x^T\}$. Our goal is to minimize the risk, and also to make the the sensitivity of parameters small. This is naturally described as an optimization problem with two objectives. To solve this bi-criterion problem, we minimize the weight sum of the objectives by introducing a $\tau > 0$, as a trade-off factor. We propose an optimal trade-off model for portfolio selection with sensitivity of parameters, i.e.,

$$\min \tau x \Sigma x^T + \max_{i \in S} \{x \Sigma_i x^T\} - \min_{i \in S} \{x \Sigma_i x^T\}$$

s.t. $\langle \mu, x \rangle \geq \rho$, $\langle e, x \rangle = 1$, $l \leq x \leq u$,  

$$(\text{TMV}_sc)$$

where $\tau \in (0, \infty)$ is a trade-off factor. Since $\Sigma_i$ is an indefinite matrix, $x \Sigma_i x^T$ is a class of non-convex quadratic functions. Due to the nonconvexity and nonsmoothness of the objective function, it is hard to solve $$(\text{TMV}_{sc}).$$

Since $\min_{i \in S} \{x \Sigma_i x^T\} = - \max_{i \in S} \{-x \Sigma_i x^T\}$, the $$(\text{TMV}_{sc})$$ can be rewritten as,

$$\min \tau x \Sigma x^T + \max_{i \in S} \{x \Sigma_i x^T\} + \max_{i \in S} \{-x \Sigma_i x^T\}$$

s.t. $\langle \mu, x \rangle \geq \rho$, $\langle e, x \rangle = 1$, $l \leq x \leq u$,  

$$(\text{TMV}_{sc1})$$

Thus, the objective function of $$(\text{TMV}_{sc1})$$ is a composite function that contains a convex quadratic function and two non-convex maximum functions. Then, we transform it equivalently to an unconstrained composite problem in Subsection 3.2.
3.2. Unconstrained optimization problem. To transform the \( \text{TMV}_{\text{sc}} \) to an unconstrained composite equivalently, we first introduce an indicator function of \( X = \{ x | (\mu, x) \geq \rho, (e, x) = 1, l \leq x \leq u \} \) given by,

\[
\mathcal{I}(x) = \begin{cases} 
0, & x \in X, \\
+\infty, & x \notin X.
\end{cases}
\]

Then \( \text{TMV}_{\text{sc}} \) can be equivalently expressed as an unconstrained optimization,

\[
\min_{x \in S} \tau x \Sigma x^T + \max_{i \in S} \{ x \Sigma_i x^T \} + \max_{i \in S} \{ -x \Sigma_i x^T \} + \mathcal{I}(x). \tag{UTMV_{sc}}
\]

It is obviously that both terms of \( \max_{i \in S} \{ x \Sigma_i x^T \} \) and \( \max_{i \in S} \{ -x \Sigma_i x^T \} \) are non-convex and non-smooth. Thus \( \text{UTMV}_{\text{sc}} \) is still difficult to solve.

To convexify \( \max_{i \in S} \{ x \Sigma_i x^T \} + \max_{i \in S} \{ -x \Sigma_i x^T \} \), we choose a sufficiently large constant \( \theta \) such that \( \Sigma_i + \theta I \) and \( -\Sigma_i + \theta I \) are positive semidefinite, where \( I \) is an \( m \times m \) identical matrix. The matrix \( \Sigma_i \) can be decomposed as \( \Sigma_i = \lambda_i \nu_i \nu_i^T - \gamma_i \omega_i \omega_i^T \), where \( \lambda_i \geq 0, -\gamma_i \leq 0 \) are the eigenvalues of \( \Sigma_i \) and \( \nu_i, \omega_i \) are the corresponding orthogonal unit eigenvectors. In particular, let

\[
\theta = \max \{ \lambda_1, \ldots, \lambda_m, \gamma_1, \ldots, \gamma_m \}.
\]

Therefore, \( \text{UTMV}_{\text{sc}} \) is equivalent to the following problem.

\[
\min \Psi(x) + \mathcal{X}(x), \tag{UTMV_{sc}}
\]

where

\[
\Psi(x) = f(x) + h(x), \quad f(x) = -2\theta(x, x), \quad h(x) = \tau x \Sigma x^T,
\]

\[
\mathcal{X}(x) = \max_{i \in S} \{ x(\Sigma_i + \theta I)x^T \} + \max_{i \in S} \{ x(-\Sigma_i + \theta I)x^T \} + \mathcal{I}(x).
\]

Thus, \( \Psi(x) \) is a non-convex differentiable function which consists of a convex function and a concave function. \( \mathcal{X}(x) \) is a proper convex with bounded domain \( x \in X \) but non-smooth function.

**Definition 3.1.** [12] Given a matrix \( A \in \mathbb{R}^{n \times n} \), its spectral norm is defined as the largest singular value of \( A \), i.e.,

\[
\| A \| = \sqrt{\text{maximum eigenvalue of } (A^H A)},
\]

where \( A^H \) denotes the conjugate transpose of \( A \).

**Lemma 3.2.** For any \( x, y \in X \), there exits a constant \( L_{\Psi} \) such that

\[
\| \nabla \Psi(y) - \nabla \Psi(x) \| \leq L_{\Psi} \| y - x \|.
\]

**Proof.** By \( \Psi(x) = \tau x \Sigma x^T - 2\theta(x, x) \), we have

\[
\| \nabla \Psi(y) - \nabla \Psi(x) \| = \| (2\tau y \Sigma - 4\theta y) - (2\tau x \Sigma - 4\theta x) \| \\
\leq 2\tau \| (y - x) \Sigma \| + 4\theta \| y - x \| \leq (2\tau \| \Sigma \| + 4\theta) \| y - x \|
\]

Let \( L_{\Psi} = 2\tau \| \Sigma \| + 4\theta \), we have

\[
\| \nabla \Psi(y) - \nabla \Psi(x) \| \leq L_{\Psi} \| y - x \|.
\]

Therefore, \( \nabla \Psi(x) \) is Lipschitz continuous. \( \square \)
PORTFOLIO SELECTION WITH SENSITIVITY OF PARAMETERS

Obviously, (UTMV\textsuperscript{sc1}) has the same form of (CP). Therefore, (UTMV\textsuperscript{sc1}) can be solved by Algorithm 2.1. However, if \(m\), the number of the assets for restricting the sensitivity of parameters, is chosen large, then solving (2) and (3) of Algorithm 2.1 may cause the computational difficulty. To overcome it, we develop a modified AG algorithm in the following section.

4. The modified AG algorithm

In this section, we first develop a modified AG algorithm to solve the problem (UTMV\textsuperscript{sc1}), which is described as Algorithm 4.1. Then we derive the convergence and the convergence rate of Algorithm 4.1.

Denote that \(P_i = \Sigma_i + \theta I\), \(Q_i = -\Sigma_i + \theta I\), \(i \in S\). The key step in Algorithm 2.1 is to solve problem (P), which can be reformulated equivalently into the following quadratic convex programming problem.

\[
\min_w \langle y, w \rangle + \frac{1}{2c} \|w - x\|^2 + \max_{i \in S} \{w^T P_i w\} + \max_{i \in S} \{w^T Q_i w\} \quad \text{s.t.} \quad x \in X. \tag{P1}\]

(P1) is a convex programming problem that can be solved by CVX software package developed by Grant et al. [10]. However, choosing \(m\) assets from \(n\) assets, (P1) has to be solved twice for two terms of sensitive of parameters at each iteration in Algorithm 2.1, which obviously increases the computational cost. To alleviate expensive computational cost associated with the different parameters, our idea is to only solve (P1) once to obtain \(x_k^{ag}\). Then the iteration \(x_k\) is obtained in terms of combination of \(x_k^{ag} - 1\) and \(x_k^{ag}\). In other words, our algorithm only requires the solution of one subproblem (P1) and the computational cost is decreased. Our algorithm is described as follows.

Algorithm 4.1. The modified AG algorithm

**Step 0.** Input a feasible solution \(x_0\) and the accuracy parameter \(\epsilon\). Let \(x_0^{ag} = x_0\), \(k = 1\), \(a_k = \frac{2}{k+1}\), \(b_k = \frac{1}{\pi c}\).

**Step 1.** Let
\[
x_k^{md} = (1 - a_k)x_{k-1}^{ag} + a_kx_{k-1}. \tag{4}\]

**Step 2.** Compute \(\nabla \Psi(x_k^{md})\), let
\[
x_k^{ag} = P(x_k^{md}, \nabla \Psi(x_k^{md}), b_k), \tag{5}\]
\[
x_k = x_k^{ag} + \frac{1}{\alpha_k} (x_k^{ag} - x_k^{ag} - 1). \tag{6}\]

**Step 3.** If \(\|G\| \leq \epsilon\), stop. Else \(k = k + 1\), go to Step 1.

Note that the subproblem (5) guarantees the feasibility of solution and its solution \(x_k^{ag}\) is used to compute the search direction \(x_k^{ag} - x_k^{ag} - 1\). From the termination criterion (1) and (5), the new termination criterion is as follows.

\[
G(x_k^{md}, \nabla \Psi(x_k^{md}), b_k) = \frac{1}{b_k} (x_k^{md} - x_k^{ag}). \tag{7}\]

The following Lemma 4.1 shows that \(x_k^{ag}\) approaches to a stationary point of (UTMV\textsuperscript{sc1}) as (7) decreasing.
Lemma 4.1. If \( \|G(x_k^m, \nabla \Psi(x_k^m), \beta_k)\| \leq \epsilon \) after \( k \) iterations, then
\[ -\nabla \Psi(x_k^{ag}) \in \partial \lambda(x_k^{ag}) + B(\frac{3}{2}\epsilon), \]
where \( B(r) := \{x \in \mathbb{R} : \|x\| \leq r\} \).

Proof. By \( \text{(5)} \) and the first-order optimal condition, we have
\[ -\Psi(x_k^{ag}) - \frac{1}{\beta_k}(x_k^{ag} - x_k^{md}) \in \partial \lambda(x_k^{ag}). \]
Thus, we obtain
\[ -\Psi(x_k^{ag}) + \Psi(x_k^{ag}) - \Psi(x_k^{md}) - \frac{1}{\beta_k}(x_k^{ag} - x_k^{md}) \in \partial \lambda(x_k^{ag}). \]
By Lemma \( \text{3.2} \) and \( \beta_k = \frac{1}{L_f} \), we have
\[
\|\Psi(x_k^{ag}) - \Psi(x_k^{md}) - \frac{1}{\beta_k}(x_k^{ag} - x_k^{md})\| \leq L_\Psi \|x_k^{ag} - x_k^{md}\| + \frac{1}{\beta_k} \|x_k^{ag} - x_k^{md}\|
\[
= \frac{3}{2} \left( \frac{1}{\beta_k} \|x_k^{ag} - x_k^{md}\| \right) \leq \frac{3}{2} \epsilon.
\]

If it satisfies \(-\nabla \Psi(x_k^{ag}) \in \partial \lambda(x_k^{ag}) + B(\frac{3}{2}\epsilon), \), \( x_k^{ag} \) is called an \( \epsilon \)-stationary point of \( \text{(UTMV_{sc})} \). In the following part, we prove the convergence and convergence rate of Algorithm 4.1.

Theorem 4.2. If \( x^* \) is an optimal solution for \( \text{(UTMV_{sc})} \) and \( \|x_k\|(k \geq 1) \) is bounded, then for any \( N \geq 1 \), there exists a constant \( M \) such that
\[ \min_{k=1, \ldots, N} \|G(x_k^m, \nabla \Psi(x_k^m), \beta_k)\|^2 \leq \frac{24L_\Psi(L_f + 2L_\Psi)}{N + 2}(\|x^*\|^2 + 2M^2). \]
Moreover, \( x_N^{ag} \) is an \( \epsilon \)-stationary point of \( \text{(UTMV_{sc})} \) at most \( N = \mathcal{O}(1/\epsilon^2) \) iterations.

Proof. By Lemma \( \text{3.2} \) and Lemma \( \text{2.4} \) we have the following inequality,
\[ -\frac{L_f}{2}\|y - x\|^2 \leq \Psi(y) - \Psi(x) - \langle \nabla \Psi(x), y - x \rangle \leq \frac{L_\Psi}{2}\|y - x\|^2. \]
Specially, for the right hand side of inequality, we have
\[ \Psi(x_k^{ag}) \leq \Psi(x_k^{md}) + \langle \nabla \Psi(x_k^{md}), x_k^{ag} - x_k^{md} \rangle + \frac{L_\Psi}{2} \|x_k^{ag} - x_k^{md}\|^2, \]
and for the left hand side of inequality, we have
\[
\Psi(x_k^{md}) - \|((1 - \alpha_k)\Psi(x_k^{ag}) + \alpha_k \Psi(x))
\]
\[ = \alpha_k [\Psi(x_k^{md} - \Psi(x)) + (1 - \alpha_k)[\Psi(x_k^{md}) - \Psi(x_k^{ag})]
\]
\[ \leq \alpha_k [\|\nabla \Psi(x_k^{md}, x_k^{md} - x) + \frac{L_f}{2} \|x_k^{md} - x\|^2]
\]
\[ + (1 - \alpha_k)[\|\nabla \Psi(x_k^{md}), x_k^{md} - x_k^{ag}\| + \frac{L_f}{2} \|x_k^{md} - x_k^{ag}\|^2]. \]
Combining \( \text{(8)} \) and \( \text{(9)} \), we have to compute \( \langle \nabla \Psi(x_k^{md}), x_k^{ag} - \alpha_k x - (1 - \alpha_k)x_k^{ag} \rangle \). Since \( x_k^{ag} \) is the optimal solution of \( \text{(5)} \), we have
\[ \langle \nabla \Psi(x_k^{md}), x_k^{ag} - x \rangle + \lambda(x_k^{ag}) \leq \lambda(x) + \frac{1}{2\beta_k} \left( \|x_k^{md} - x\|^2 - \|x_k^{ag} - x_k^{md}\|^2 \right), \]
for any \( x \in \mathbb{R}^n \). Let \( x = x_{k-1}^{ag} \),

\[
\langle \nabla \psi(x^m), x_{k-1}^{ag} - x_{k-1}^{ag} \rangle
\leq -\psi(x_{k-1}^{ag}) + \psi(x_{k-1}^{ag}) + \frac{1}{2\beta_k} \left[ \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 - \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 \right].
\]  

(11)

Multiplying (10) by \( \alpha_k \) and (11) by \( (1 - \alpha_k) \), respectively, we obtain

\[
(1 - \alpha_k)\langle \nabla \psi(x^m), x_{k-1}^{ag} - x_{k-1}^{ag} \rangle + \alpha_k \langle \nabla \psi(x^m), x_{k-1}^{ag} - x \rangle
\leq -\psi(x_{k-1}^{ag}) + \alpha_k \psi(x) + (1 - \alpha_k)\psi(x_{k-1}^{ag}) + \frac{1}{2\beta_k} \left[ \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 - \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 \right]
\]

(12)

Then denoting \( \Phi(x) = \psi(x) + \psi(x) \) and combining (8), (9) and (12), we have

\[
\Phi(x_{k-1}^{ag}) \leq (1 - \alpha_k)\Phi(x_{k-1}^{ag}) + \alpha_k \Phi(x) + \frac{\alpha_k(L_f \beta_k + 1)}{2\beta_k} \|x_{k-1}^{ag} - x\|^2
\]

\[
+ \frac{1 - \alpha_k}{2\beta_k} \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 + \frac{L_q \beta_k - 1}{2\beta_k} \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2
\]

\[
= (1 - \alpha_k)\Phi(x_{k-1}^{ag}) + \alpha_k \Phi(x) + \frac{\alpha_k(L_f \beta_k + 1)}{2\beta_k} \|x_{k-1}^{ag} - x\|^2
\]

\[
+ \frac{\alpha_k^2(1 - \alpha_k)(L_f \beta_k + 1)}{2\beta_k} \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 + \frac{L_q \beta_k - 1}{2\beta_k} \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2,
\]

where the last equality follows form the equation (11). Subtracting \( \Phi(x) \) from both side of the above inequality, re-arranging the terms, we obtain

\[
\phi(x_{k-1}^{ag}) - \phi(x) \leq \sum_{k=1}^{N} \frac{1 - \alpha_k}{\Gamma_k} \|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2
\]

(13)

\[
\leq \sum_{k=1}^{N} \frac{\alpha_k(L_f \beta_k + 1)}{2\beta_k \Gamma_k} \left[ \|x_{k-1}^{ag} - x\|^2 + \alpha_k(1 - \alpha_k)\|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2 \right],
\]

where

\[
\Gamma_k := \begin{cases} 
1, & k = 1, \\
(1 - \alpha_k)\Gamma_{k-1}, & k \geq 2.
\end{cases}
\]  

(14)

Observing Lemma 2.5, there exists a constant \( M_1 \) such that \( \|x_{k-1}^{ag}\| \leq M_1 \) and \( x_{k-1}^{ag} \in X \). Moreover, by the presume that \( \|x_{k-1}^{ag}\| \) is bounded, there also exists a constant \( M_2 \) such that \( \|x_{k-1}^{ag}\| \leq M_2 \). Therefore, letting \( M = \max\{M_1, M_2\} \), \( x = x^* \), and using Jensen’s inequality for \( \| \cdot \| \), we have

\[
\|x_{k-1}^{ag} - x^*\|^2 + \alpha_k(1 - \alpha_k)(\|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2)
\]

\[
\leq 2(\|x^*\|^2 + \|x_{k-1}^{ag}\|^2 + \alpha_k(1 - \alpha_k)(\|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2) + \|x_{k-1}^{ag}\|^2)
\]

\[
\leq 2(\|x^*\|^2 + \alpha_k(\|x_{k-1}^{ag}\|^2 + \|x_{k-1}^{ag}\|^2) + \alpha_k(1 - \alpha_k)(\|x_{k-1}^{ag} - x_{k-1}^{ag}\|^2) + \|x_{k-1}^{ag}\|^2)
\]

\[
\leq 2(\|x^*\|^2 + 2M^2).
\]
Replacing the above result in (13), we obtain
\[
\frac{\Phi(x_{N}^{ag}) - \Phi(x^*)}{\Gamma_N} = \sum_{k=1}^{N} \frac{1 - L_{\Psi}\beta_k}{2\beta_k\Gamma_k} \|x_k^{ag} - x_k^{md}\|^2 \leq \sum_{k=1}^{N} \frac{\alpha_k(L_f\beta_k + 1)}{\beta_k\Gamma_k} = \frac{L_f + 2L_{\Psi}}{\Gamma_N} (\|x^*\|^2 + 2M^2),
\]
where \(\beta_k = \frac{1}{2\Psi}\) and the last inequality follows that
\[
\sum_{k=1}^{N} \frac{\alpha_k}{\Gamma_k} = \frac{1}{\Gamma_1} + \sum_{n=1}^{N} \frac{1}{\Gamma_k} (1 - \frac{\Gamma_k}{\Gamma_{k-1}}) = \frac{1}{\Gamma_1} + \sum_{k=2}^{N} \frac{1}{\Gamma_{k-1}} = \frac{1}{\Gamma_N}.
\]
For the fact of \(\Phi(x_{N}^{ag}) - \Phi(x^*) \geq 0\), we obtain
\[
\frac{1}{16L_{\Psi}} \sum_{k=1}^{N} k(k+1) \|G(x_k^{md}, \nabla \Psi(x_k^{md}), \beta_k)\|^2 \leq \sum_{k=1}^{N} \frac{1 - L_{\Psi}\beta_k}{2\beta_k\Gamma_k} \|x_k^{ag} - x_k^{md}\|^2 \leq \frac{L_f + 2L_{\Psi}}{\Gamma_N} (\|x^*\|^2 + 2M^2).
\]
From \(\alpha_k = \frac{2}{k+1}\) and (14), we have
\[
\Gamma_k = \frac{2}{k(k+1)}.
\]
Therefore, substituting \(\Gamma_N\) in the above equation, we have
\[
\min_{k=1,\ldots,N} \|G(x_k^{md}, \nabla \Psi(x_k^{md}), \beta_k)\|^2 \leq \frac{24L_{\Psi}(L_f + 2L_{\Psi})}{N+2} (\|x^*\|^2 + 2M^2).
\]
Therefore, in view of Lemma 4.1, \(x_{N}^{ag}\) is an \(\epsilon\)-stationary point of (UTMV\(_{sc}\)) at most \(N = O(1/\epsilon^2)\) iterations.

5. Empirical analysis. In this section, we take the empirical analysis for our proposed optimal trade-off model compared with the MV model. We first describe a stress test problem using data from Standard & Poor’s 500 (S&P 500) and compare the efficient frontiers and the ratios of mean to standard deviation of the portfolios between our model and the MV model. Then we analyse the out-of-sample performances of the portfolios using Monte Carlo simulation. Finally, we explore the region of the parameter \(\tau\) in our model. Let \(l_i = 0, u_i = 0.3, i = 1,\ldots, 33, m = n = 33\), and \(\epsilon = 10^{-4}\). The proposed model is solved by the modified AG algorithm which is implemented in Matlab R2012(a) win64-bit on a PC with 2.30GHZ CPU processor.

5.1. Sample analysis. We consider a portfolio selection problem with S&P 500 historical data from the period of July 24, 2006 to February 24, 2012. Weekly rates of return are used to estimate their mean \(\mu\), standard deviation \(\sigma\) and the covariant matrix \(\Sigma\). We select 33 stocks according to the statistic analysis in [13, 3], listed in Table I.

To compare the efficient frontier of the (TMV\(_{sc}\)) model and the MV model, we first observe the efficient frontiers without stress testing. Set the trade-off factor \(\tau = 1, 5, 10, 20\), and the expected return \(\rho \in [7 \times 10^{-3}, 11 \times 10^{-3}]\). For each value of \(\rho\), we obtain the optimal risk by performing the (TMV\(_{sc}\)) model and the MV model. The efficient frontier is composed of investments with the highest return and the
Table 1. 33 stocks from S&P 500

| Stock | Name                           | Stock | Name                           |
|-------|--------------------------------|-------|--------------------------------|
| 1     | MASTERCARD                     | 18    | MEDCO HEALTH SLTN.             |
| 2     | PRICELINE.COM                  | 19    | SOUTHWESTERN ENERGY           |
| 3     | MCDONALDS.COM                  | 20    | CVS CAREMARK                  |
| 4     | AUTOZONE                       | 21    | J M SMUCKER                   |
| 5     | WATSON PHARMS.                 | 22    | URBAN OUTFITTERS             |
| 6     | FAMILY DOLLAR STORES           | 23    | APOLLO GP.'A'                 |
| 7     | PERRIGO                        | 24    | CELGENE                       |
| 8     | STERICYCLE                     | 25    | INTERCONTINENTAL EX.          |
| 9     | INTUITIVE SURGICAL             | 26    | LOCKHEED MARTIN               |
| 10    | EDWARDS LIFESCIENCES           | 27    | ALTRIA GROUP                  |
| 11    | GOODRICH                       | 28    | HORMEL FOODS                  |
| 12    | FIDELITY NAT.INFO.SVS.         | 29    | NETFLIX                       |
| 13    | F5 NETWORKS                    | 30    | MICRON TECHNOLOGY             |
| 14    | WAL MART STORES                | 31    | VARIAN MED.SYS.               |
| 15    | COCA COLA                      | 32    | BROWN-FORMAN ‘B’              |
| 16    | BIOGEN IDEC                    | 33    | GAMESTOP ‘A’                  |
| 17    | TRAVELERS COS.                 |       |                                |

Figure 1. Efficient frontiers of (TMVsc) and (MV)

The figure illustrates that the efficient frontiers of the (TMVsc) model are below the efficient frontier of the MV model. As the value of \( \tau \) increasing, the efficient frontier of the (TMVsc) model moves downward.

Then we carry out the stress test and suppose that the rates of return and the standard deviations under stress scenario are:

\[
\hat{\mu}_i = 0.7\mu_i, \quad \hat{\sigma}_i = 1.3\sigma_i, \quad i = 1, \ldots, 7,
\]
where $i$ denotes seven assets with largest sensitivity of parameters. Moreover, we reconstruct the MV model with new $\hat{\mu}$ and $\hat{\sigma}$. The portfolios of the reconstructed MV model are regarded as the optimal portfolios. Similar to Figure 1, the efficient frontiers under stress scenario are obtained. As shown in Figure 2, the efficient frontiers of the (TMV$_{sc}$) model are above the efficient frontier of the MV model. We also observe that the efficient frontier of the (TMV$_{sc}$) model is close to the optimal one for $\tau = 5$.

We further compute the ratios of mean to standard deviation of the portfolios. As the numerical results in Table 2, for the case of $\tau = 5$ and $\tau = 10$, the ratios of mean to standard deviation of the (TMV$_{sc}$) model are greater than the ratios of the MV model. Moreover, the ratio of mean to standard deviation of the (TMV$_{sc}$) model are even greater than the optimal one when $\rho = 9 \times 10^{-3}$ and $\tau = 5$. As to the criterion of ratio of mean to standard deviation, the (TMV$_{sc}$) model performs better than the MV model.

5.2. Out-of-sample analysis. We now compare the out-of-sample performances of the portfolios generated by the (TMV$_{sc}$) model and the MV model.

In particular, the weekly rates of return are generated by Monte Carlo simulation with 10000 samples from the normal distribution $N(\hat{\mu}_i, \hat{\sigma}_i^2)$. Using the weekly data, we compute the return and the risk of portfolios in Subsection 5.1. The ratios of the mean to the standard deviation are obtained and listed in Table 3.
As illustrated in Table 3, the ratios of mean to standard deviation of the \((\text{TMV}_{sc})\) model are greater than the ratios of the MV model for \(\tau = 5\). Moreover, the ratios of mean to standard deviation of the \((\text{TMV}_{sc})\) model are greater than the optimal one for the case of \(\rho = 8 \times 10^{-3}\) and \(\rho = 9 \times 10^{-3}\). Fixed \(\rho = 8 \times 10^{-3}\), we compare the accumulated returns of our model and the MV model in Figure 3.

Obviously, the accumulated return of our model is upward that of the MV model. The empirical analysis illustrates that the \((\text{TMV}_{sc})\) model makes sense for a stable return and a low risk.

5.3. Discussion of \(\tau\). The parameter \(\tau\) balances the risk and sensitivity of parameters in the \((\text{TMV}_{sc})\) model. To explore the region of \(\tau\), we preform the \((\text{TMV}_{sc})\) model and compare both optimal value of risk and that of the sensitivity of parameters, respectively, based on data from S&P 500 and Hongkong (HK) stocks. Obviously, the larger the value of \(\tau\) is, the smaller the value of the risk \(x^\Sigma x^T\) is. Moreover, we compare the rate of the risk, defined by \(\text{Rate}_R = \frac{\text{Risk}(\text{TMV}_{sc})}{\text{Risk}(\text{MV})} - 1\), and the rate of the sensitivity of parameters defined by \(\text{Rate}_S = |\frac{\text{Sensitivity}(\text{TMV}_{sc})}{\text{Sensitivity}(\text{MV})} - 1|\), respectively. The numerical results are listed and illustrated in Table 4. Fixed \(\rho = 7 \times 10^{-3}\) and \(\tau = 5\), the risk of the \((\text{TMV}_{sc})\) model is \(0.4902 \times 10^{-3}\). In contrast to the risk of the MV model, there is almost 20% increasing. Similarly, the sensitivity of the \((\text{TMV}_{sc})\) model is \(1.7136 \times 10^{-3}\). It is almost 57% decreasing, compared with that of the MV...
model. Let the value of $\tau$ increase as $\tau = 10$, $\tau = 15$ and $\tau = 20$, respectively. We observe that the values of $Rate_{R}$ and $Rate_{S}$ go to less, respectively. Summarily, we conclude that $\tau$ makes sense to balance the risk and the sensitivity of parameters in the \((TMV_{sc})\) model with a range $\tau \in [5, 20]$.

To further verify the property of $\tau$ and its region, we perform the \((TMV_{sc})\) model again by new data of HK stocks to compare the risk, the sensitivity of parameters, $Rate_{R}$ and $Rate_{S}$, respectively. The corresponding numerical results are recorded in Table 5. As showed in Table 5, fixed $\rho = 3 \times 10^{-3}$ and $\tau = 10$, the risk of the \((TMV_{sc})\) model is $0.2819 \times 10^{-3}$, greater almost 14% than that of the MV model. The sensitivity of the \((TMV_{sc})\) model is $4.8713 \times 10^{-3}$, less almost 38% than that of the MV model. We observe the similar conclusion to the previously noted numerical results in Table 4 with a range $\tau \in [20, 40]$. Moreover, we find that the region of $\tau$ depends on the different data.

6. Numerical experiments. In this section, we first compare the CPU time and iterations between the modified algorithm and Algorithm 2.1, respectively, based on the data from S&P 500. We then present the computational results of Algorithm 4.1 based on randomly generated data. The parameters in Algorithm 2.1 are set as $\alpha_k = \frac{2}{k+1}$, $\beta_k = \frac{1}{2L}\psi$, and $\lambda_k = \frac{k\beta_k}{2}$. The algorithms are implemented in Matlab R2012(a) win64-bit on a PC with 2.30GHZ CPU processor.

6.1. Numerical experiments for S&P 500. We first perform the \((TMV_{sc})\) model by Algorithm 2.1 and Algorithm 4.1 and compare the CPU time and iterations of algorithms, respectively, based on the data from S&P 500.

For each instance, the optimal value of the model \((TMV_{sc})\) solved by Algorithm 4.1 is denoted as "Opt". Meanwhile, Algorithm 2.1 is terminated as the objective
Table 5. Discussion of $\tau$ for HK stocks

| $\rho$ $(\times 10^{-3})$ | $\tau$ | Risk $(\times 10^{-3})$ | Rate$_R$ (%) | Sensitivity $(\times 10^{-3})$ | Rate$_S$ (%) |
|--------------------------|-------|------------------------|-------------|----------------|-------------|
| MV                       | 0.2469| -                      | 7.9021      | -              | -           |
| $\tau = 10$              | 0.2819| 14.18                  | 4.8713      | 38.35          |
| $\tau = 20$              | 0.2616| 5.95                   | 5.4896      | 30.53          |
| $\tau = 30$              | 0.2595| 5.10                   | 5.6577      | 28.40          |
| $\tau = 40$              | 0.2582| 4.58                   | 5.4896      | 26.64          |
| 3                        |       |                        |             |                |
| MV                       | 0.3118| -                      | 8.2937      | -              |
| $\tau = 10$              | 0.3505| 12.41                  | 5.386       | 35.06          |
| $\tau = 20$              | 0.3356| 7.63                   | 5.6569      | 31.79          |
| $\tau = 30$              | 0.3308| 6.09                   | 5.9154      | 28.68          |
| $\tau = 40$              | 0.3272| 4.94                   | 6.1602      | 25.72          |
| 3.5                      |       |                        |             |                |
| MV                       | 0.4445| -                      | 9.7782      | -              |
| $\tau = 10$              | 0.4500| 1.24                   | 9.1207      | 6.72           |
| $\tau = 20$              | 0.4482| 0.83                   | 9.2027      | 5.89           |
| $\tau = 30$              | 0.4472| 0.61                   | 9.2756      | 5.14           |
| $\tau = 40$              | 0.4462| 0.40                   | 9.3756      | 4.43           |

Table 6. Comparison Algorithm 4.1 and Algorithm 2.1

| $\tau$ | $\rho$ $(\times 10^{-3})$ | Opt | Algorithm 4.1 | Algorithm 2.1 |
|--------|--------------------------|-----|---------------|---------------|
|        |                          | CPU time | $N_{iter}$ | CPU time | $N_{iter}$ |
| 5      | 5.5                      | 0.0171  | 96.9  | 58 | 670.6 | 214 |
| 5      | 6                       | 0.0170  | 151.0 | 88 | $\geq 1000$ | $\geq 279$ |
| 5      | 7                       | 0.0304  | 142.5 | 85 | $\geq 1000$ | $\geq 300$ |
| 10     | 5.5                      | 0.0297  | 104.2 | 62 | 279.0 | 82 |
| 10     | 6                       | 0.0312  | 123.8 | 75 | 302.2 | 88 |
| 10     | 7                       | 0.0424  | 80.2  | 43 | $\geq 1000$ | $\geq 327$ |
| 20     | 5.5                      | 0.0351  | 173.0 | 102 | 627.1 | 186 |
| 20     | 6                       | 0.0366  | 141.0 | 84 | 350.5 | 105 |
| 20     | 7                       | 0.0494  | 92.1  | 52 | $\geq 1000$ | $\geq 350$ |

The CPU time and iterations are listed in Table 6 respectively. Where the iteration is denoted as $N_{iter}$.

It is obviously that the CPU time in Algorithm 2.1 are even more than 1000 seconds and iterations are more than 350, while all the CPU time of Algorithm 4.1 are less than 200 seconds and iterations are less than 110. Therefore, Algorithm 4.1 solves the model (TMV$_{sc}$) more efficiently than Algorithm 2.1. To further analyse the effectiveness of Algorithm 4.1 on (TMV$_{sc}$), we test it with different $m$ and $n$.

In the following implementation, $\tau = 5$, $\rho = 4 \times 10^{-3}$ and the termination criterion $\epsilon = 10^{-4}$. For each pair of $m$ and $n$, we generate five instances by selecting value is not more than Opt. The CPU time and iterations are listed in Table 6 respectively. Where the iteration is denoted as $N_{iter}$.
Table 7. Numerical experiments for S&P 500

| n  | m  | min | max  | average | N_{iter} |
|----|----|-----|------|---------|----------|
| 10 | 5  | 8.2 | 27.5 | 17.6    | 33       |
| 10 | 10 | 11.5| 36.0 | 24.4    | 32       |
| 20 | 5  | 15.5| 44.8 | 34.1    | 62       |
| 20 | 10 | 34.6| 34.9 | 42.6    | 56       |
| 30 | 10 | 23.4| 95.5 | 62.1    | 78       |
| 30 | 20 | 70.0| 125.3|104.1    | 89       |
| 40 | 10 | 51.9|100.1 |68.4     | 84       |
| 40 | 20 | 74.8|123.5 |85.3     | 68       |
| 50 | 10 | 41.8| 59.4 | 52.0    | 64       |
| 50 | 20 | 76.3|123.0 |95.5     | 82       |
| 100| 10 | 39.6|128.1 |82.0     | 79       |
| 100| 20 | 85.8|277.5 |156.7    | 71       |
| 100| 50 | 615.8|830.8 |738.2    | 185      |
| 200| 20 | 225.5|530.3 |405.3    | 137      |
| 200| 50 | 722.9|1106.8|911.8    | 165      |

For the rest parts of numerical experiments, we mainly report the minimum, maximum, average CPU time and average iterations. According to the numerical results in Table 7, when \( n \leq 100 \) and \( m \) is fixed, we find that the CPU time are relative stable. However, the CPU time is almost double when \( m \) doubled. It means that the CPU time of Algorithm 4.1 is more sensitive to \( m \) than \( n \).

6.2. Numerical experiments for randomly data. We further present the numerical results of Algorithm 4.1 in the instances generated randomly by the same method mentioned in [27]. In particular, the instances are generated by the single factor model as followed.

\[
    r_i = \alpha_i + \beta_i r_m + e_i, \quad i = 1, \ldots, n, \]

where \( r_i \) is the return of asset \( i \), \( r_m \) is the return of market index and \( e_i \) is the residual return of asset \( i \). Therefore, we have \( \mu_i = \alpha_i + \beta_i E(r_m), \sigma_{ii} = \beta_i^2 \text{Var}(r_m) + \text{Var}(e_i) \) and \( \sigma_{ij} = \beta_i \beta_j \text{Var}(r_m) \), where the parameters are set as followed.

- \( \alpha_i = 10^{-5} \times r_d, i = 1, \ldots, n \), where \( r_d \) is a random number generated by the standard normal distribution.
- \( \beta_i \in [0.6, 1.4], i = 1, \ldots, n \) is randomly generated by the uniform distribution.
- \( E(r_m) = 4 \times 10^{-3}, \text{Var}(r_m) = 0.03 \), and \( \text{Var}(e_i) \in [0, 0.2 \times 10^{-3}], i = 1, \ldots, n \) is randomly generated by the uniform distribution.
- \( \tau = 5, \rho = 4 \times 10^{-3} \).

For each instance generated, the computational results are listed in Table 8. It has been seen that Algorithm 4.1 is sensitive to \( m \). Moreover, given fixed \( m \), if \( n \) increases, then the CPU time decreases. This observation shows that the CPU time depends on the different value of \( m/n \). Given \( m \in [5, 50] \), we illustrate the behavior

\[
    n \text{ stocks randomly from S&P 500 to obtain the computational results in Table 7.} \]
Table 8. Numerical experiments for randomly data

| n  | m  | min | max | average | N_{iter} |
|----|----|-----|-----|---------|----------|
| 10 | 5  | 12.8| 14.5| 13.3    | 25       |
| 10 | 10 | 36.7| 37.4| 36.9    | 51       |
| 20 | 5  | 7.1 | 7.4 | 7.3     | 14       |
| 20 | 10 | 14.6| 34.3| 24.8    | 32       |
| 30 | 10 | 13.2| 15.5| 14.5    | 20       |
| 30 | 20 | 45.4| 46.9| 46.1    | 39       |
| 40 | 10 | 9.4 | 10.0| 9.7     | 13       |
| 40 | 20 | 22.7| 26.1| 24.0    | 19       |
| 50 | 10 | 8.6 | 10.4| 9.5     | 13       |
| 50 | 20 | 22.3| 24.2| 23.1    | 19       |
| 100| 10 | 5.8 | 12.8| 9.3     | 10       |
| 100| 20 | 16.6| 28.8| 22.6    | 13       |
| 100| 50 | 47.1| 54.9| 51.5    | 15       |
| 200| 20 | 33.3| 39.6| 36.5    | 17       |
| 200| 50 | 50.9| 78.1| 66.7    | 14       |

Figure 4. Performance of the CPU time for different value of $m/n$

of CPU time in Figure 4, where the lines with “○, □, ×” denote the CPU time with $m/n = 1, 1/2, 1/3$, respectively.

7. Conclusions. In this paper, we have proposed an optimal trade-off model for portfolio selection with sensitivity of parameters. Furthermore, we have developed a modified AG algorithm to solve the optimal trade-off model. The convergence and the convergence rate of our algorithm have been derived. The empirical analysis has indicated that the optimal trade-off model results in a stable return with lower
risk under the stress test. The numerical experiments have been demonstrated that the modified AG algorithm outperforms the existed AG algorithm in both the CPU time and the iterations, respectively.

REFERENCES

[1] F. A. Al-Khayyal, C. Larsen and T. V. Voorhis, A relaxation method for nonconvex quadratically constrained programs, *Journal of Global Optimization*, 6 (1995), 215–230.

[2] C. Audet, P. Hansen, B. Jaumard and G. Savard, A branch-and-cut algorithm for nonconvex quadratically constrained quadratic programming, *Mathematical Programming*, 87 (2000), 131–152.

[3] V. Boginski, S. Butenko and P. M. Pardalos, Statistical analysis of financial networks, *Computational Statistics & Data Analysis*, 48 (2005), 431–443.

[4] V. K. Chopra and W. T. Ziemba, The effect of errors in means, variances, and covariances on optimal portfolio choice, *Journal of Portfolio Management*, 19 (1993), 6–11.

[5] X. T. Cui, X. L. Sun and D. Sha, An empirical study on discrete optimization models for portfolio selection, *Journal of Industrial and Management Optimization*, 5 (2009), 33–46.

[6] X. T. Cui, Studies on Portfolio Selection Problems with Different Risk Measures and Trading Constraints, O224, Fudan University, 2013.

[7] Z. B. Deng, Y. Q. Bai, S. C. Fang, T. Ye and W. X. Xing, A branch-and-cut approach to portfolio selection with marginal risk control in a linear cone programming framework, *Journal of Systems Science and Systems Engineering*, 22 (2013), 385–400.

[8] S. Ghadimi and G. H. Lan, Accelerated gradient methods for nonconvex nonlinear and stochastic programming (accepted), *Mathematical Programming*, (2015). arXiv:1310.3787v1

[9] D. Goldfarb and G. Iyengar, Robust portfolio selection problems, *Mathematics of Operations Research*, 28 (2000), 1–38.

[10] M. Grant and S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, http://cvxr.com/cvx/ 2014.

[11] P. Horst, P. M. Pardolos and N. V. Thoai, *Introduction to Global Optimization*, Kluwer Academic Publishers, 1995.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.

[13] V. Kalyagin, A. Koldanov, P. Koldanov and V. Zamaraev, Market graph and markowitz model, *Optimization in Science and Engineering: In Honor of the 60th Birthday of Panos M Pardalos*, Springer Science (2014), 293–306.

[14] G. H. Lan, An optimal method for stochastic composite optimization, *Mathematical Programming*, 133 (2012), 365–397.

[15] Q. Li and Y. Q. Bai, Optimal trade-off portfolio selection between total risk and maximum relative marginal risk (accepted) *Optimization Methods & Software*, 31 (2016), 681–700.

[16] J. Lindereth, A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs, *Mathematical Programming*, 103 (2005), 251–282.

[17] H. M. Markowitz, Portfolio selection, *Journal of Finance*, 7 (1952), 77–91.

[18] Y. E. Nesterov, A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$, *Doklady AN SSSR*, 269 (1983), 543–547.

[19] Y. E. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer, 2003.

[20] Y. E. Nesterov, Smooth minimization of non-smooth functions, *Mathematical Programming*, 103 (2005), 127–152.

[21] U. Raber, A simplicial branch-and-bound method for solving nonconvex all-quadratic programs, *Journal of Global Optimization*, 13 (1998), 417–432.

[22] B. Scherer, Can robust portfolio optimization help to build better portfolios? *Journal of Asset Management*, 7 (2007), 374–387.

[23] X. L. Sun, X. J. Zheng and D. Li, Recent advances in mathematical programming with semi-continuous variables and cardinality constraint, *Journal of the Operations Research Society of China*, 1 (2013), 55–77.

[24] Y. F. Sun, A. Grace, K. L. Teo and G. L. Zhou, Portfolio optimization using a new probabilistic risk measure, *Journal of Industrial and Management Optimization*, 11 (2015), 1275–1283.

[25] K. L. Teo and X. Q. Yang, Portfolio selection problem with minimax type risk function, *Annals of Operations Research*, 101 (2001), 333–349.
[26] Y. Tian, S. C. Fang, Z. B. Deng and Q. W. Jin, "Cardinality constrained portfolio selection problem: A completely positive programming approach," *Journal of Industrial and Management Optimization*, 12 (2016), 1041–1056.

[27] S. S. Zhu, D. Li and X. L. Sun, "Portfolio selection with marginal risk control," *The Journal of Computational Finance*, 14 (2010), 3–28.

Received October 2015; revised March 2016.

E-mail address: yqbai@shu.edu.cn
E-mail address: weiyudan@126.com
E-mail address: liqian15123329166@163.com