Original Research Article

PENALTY ALGORITHM BAESD ON THREE-TERM CONJUGATE GRADIENT METHOD FOR UNCONSTRAINED OPTIMIZATION PORTFOLIO MANAGEMENT PROBLEM

ABSTRACT

In solving unconstrained optimization problems, the conjugate gradient method is regarded as one of the most powerful approaches due to its smaller storage requirements and computation cost. Then, a class of penalty algorithms based on three-term conjugate gradient methods was developed to find the solution of an unconstrained minimization portfolio management problem, where the objective function is a piecewise quadratic polynomial. By implementing the proposed algorithm to solve some selected unconstrained optimization problems, resulted in improvement in the total number of iterations and CPU time. It was shown that this algorithm is promising

Keywords: Three-term conjugate gradient, portfolio management, unconstrained optimization, and biobjectives optimization.

1 Introduction

Portfolio management problem deals with allocating ones assets among several securities to maximize the return of assets and to minimize the investment risk, Markowitzs mean variance model(1), and the solution of his mean-variance methodology has been the center of the consequent research activities and forms the basis for the development of modern portfolio management theory. Commonly, the portfolio management problem has the following mathematical description. Assume that there are n kinds of securities. The return rate of the kth security is denoted as R_k, $k = 1, 2, 3, ..., n$, Let x_k be the proportion of total assets devoted to the kth security,

$$\sum_{k=1}^{n} x_k \tag{1.1}$$

In the real setting, due to uncertainty, the return rates R_k, $k = 1, 2, 3, ..., n$ are random parameters. Hence, the total return of the assets

$$R(x) = \sum_{k=1}^{n} R_k x_k \tag{1.2}$$

is also random. In this situation, the risk of investment has to be taken into consideration. In the classical model, this risk is measured by the variance of

$$b \geq x \geq a$$

where $\eta \in R^T$ is the expected value vector of R, and a, b $\in R^T$are two given vectors denoting the lower and the upper bounds of decision vector, respectively.

Obviously, if $\lambda = 0$ in (1.5), then it implies that the return is maximized regardless of the investment risk. On the other hand, if $\lambda = 0$ then the risk is minimized without consideration on the investment income. Increasing value of λ in the interval$[0, 1]$ indicates an increasingly weight of the invest risk, and vice versa.

For a fixed $\lambda \in (0, 1)$, it is noted that (1.5) is a quadratic programming problem. Since it has been shown that the matrix V is positive semidefinite, the problem (1.5) is a convex quadratic programming (CQP). For a CQP, there exist a lot of efficient methods to find its minimizers. Among them, active-set methods, interior-point methods, and gradient projection methods have been widely used since the 1970s. For their detailed numerical performances, one can see (6-9) and the references therein. However, the efficiency of those methods seriously depends on the factorization techniques of matrix at each iteration, often exploiting the sparsity in V for a large-scale quadratic programming. So, from the viewpoint of smaller storage requirements and computation cost, the methods mentioned above must not be most suitable for solving the problem (1.5) if V is a dense matrix.
Fortunately, recent research shows that the Three-term conjugate gradient methods can remedy the drawback in factorization of Hessian matrix for an unconstrained minimization problem. At each iteration, it is only involved with computing the gradient of objective function. For details in this direction, see, for example,(10-18). Motivated by the advantage of the conjugate gradient methods, the first aim of this paper is to reformulate problem (1.5) as an equivalent unconstrained optimization problem. Then, we are going to develop an efficient algorithm based on three-term conjugate gradient methods to find its solution. The effectiveness of such algorithm will be tested by implementing the designed algorithm to solve some real problems from CUTEr Suite.
The lay out of the paper is as follows. Section 2 is devoted to the reformulation of the original constrained problem. Some features of the subproblem will be presented. Then, in Section 3, we are going to develop a penalty algorithm based on Three-term conjugate gradient methods. Section 4 will provide applications of the proposed algorithm. The last section concludes with some final remarks.

2 REFORMATION

Firstly, for brevity, denote

$$c = (c_j)_{nx1} = -(1 - \lambda)\eta, \qquad Q = (q_{ij})_{nxn} = 2\lambda V$$

Then, the problem (4) reads

$$\begin{aligned} minimize f(x) = c^T x + \frac{1}{2} x^T Q x \\ subject \;\; to \;\; e^T x = 1 \qquad (2.1) \\ a \leq x \leq b \end{aligned}$$

Since the covariance matrix V is symmetric positive semidefinite, Q also has such property. Thus, $f(x)$ is a convex function.
For the equality constraint $e^T x = 1$ and the inequality constraints $a \leq x \leq b$, we define a

function $P : R^{n+1} \to R$ which is used to describe the constraints violation

$$P(x;\theta) = \frac{\theta}{2}[(e^T x - 1)^2 + ||min(x-a,0)||^2 + ||min(b-x,0)||^2 \quad (2.2)$$

where$\theta = 0$ is called penalty parameter, and $||.||$ denotes the 2-norm of vector. If x is a feasible point of problem (1.5), then

$$P(x;\theta) = 0 \quad (2.3)$$

Actually, the larger the absolute value of $P(x;\theta) = 0$ is, the further x from the feasible region is. The function $F : R^{n+1} \to R$

$$F(x;\theta) = c^T x + \frac{1}{2}x^T Qx + \frac{\theta}{2}[(e^T x - 1)^2 + ||min(x-a,0)||^2 + ||min(b-x,0)||^2 \quad (2.4)$$

is said to be a penalty function of the problem (2.1) It is noted that F has the following features:
1. F is a piecewise quadratic polynomial
2. F is piecewise continuously differentiable.
3. If Q is positive semidefinite, then F is a piecewise convex quadratic function.

3 Penalty Algorithm Based on Three-term Conjugate Gradient Method

Among all methods for the unconstrained optimization problems, the conjugate gradient method is regarded as one of the most powerful approaches due to its smaller storage requirements and computation cost. Its priorities over other methods have been addressed in many literatures. For example, in (19-21) the global convergence theory and the detailed numerical performances on the conjugate gradient methods have been extensively investigated. Since the number of the possible selected securities in the investment management is large and the matrix $Q(x;\theta)$ may be dense, it is natura that the conjugate gradient method is selected to find the minimizer of F for some given θ. However. The standard procedures of minimizing a quadratic function can not be directly employed. To develop a new algorithm, we first propose an rule of updating the coefficients in F.
Regarding the coefficients of the quadratic terms in

$$\frac{\theta}{2}[(e^T x - 1)^2 + ||min(x-a,0)||^2 + ||min(b-x,0)||^2 \quad (3.1)$$

we modify $Q = q_{ij}$ according to the following update rule:

$$q_{ij} = \begin{cases} q_{ij} & i = j \\ q_{ii} & i = j \quad a_i \leq x_i \leq b_i \\ (q_{ij} + \theta) & i = j \quad a_i > x_i \quad or \quad b_i > x_i \end{cases} \quad (3.2)$$

Regarding the coefficients of the linear terms in

$$\frac{\theta}{2}[(e^T x - 1)^2 + ||min(x-a,0)||^2 + ||min(b-x,0)||^2 \quad (3.3)$$

we modify $c = c_{ij}$ according to the following update rule:

$$c_{ij} = \begin{cases} c_i & a_i \leq x_i \leq b_i \\ c_i - \theta a_i & if \quad x_i < a_i \\ c_i - \theta b_i & if \quad x_i > b_i \end{cases} \tag{3.4}$$

Define

$$\bar{Q} = Q + \theta ee^T \qquad \bar{c} = c - \theta e \tag{3.5}$$

The conjugate gradient method will be employed into an ordinary minimization of quadratic function:

$$minimize f(x) = c_0 + \bar{c}^T x + \frac{1}{2} x^T \bar{Q} x \tag{3.6}$$

where θ is a given parameter. It is easy to see that

$$g_x = \bar{Q}x + \bar{c} \tag{3.7}$$

Although there exist several variations on the conjugate gradient method.

The success of the three-term conjugate method, heavily depends on the choice of the step length α_k and search direction d_k i.e,the different choices of the step length and search direction lead to different convergence properties.

There are three ways to determine the value of the step length, namely, the exact line search, the inexact line search and use of formula. More often, it is impracticable to use the exact line search. The use of formula is a recent development which efficiency is still being investigated.

Among the several inexact line searches available, the Armijo rule is adjudged as one of the most useful and the easiest implementable procedure, Shi (22). The line search can be described as follows:
Given $s > 0$, $\delta \in (0, 1)$ and $\sigma \in (0, 1)$ find $\alpha_k = max\{s, s\delta, s\delta^2, \ldots\}$ such that

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\sigma \alpha_k g_k^T d_k, \quad k = 0, 1, 2, ..., n. \tag{3.8}$$

One requirement of the search direction d_i is the satisfaction of descent condition to guarantee the attainment of the minimum value of the objective function $f(x)$. The CG method easily satisfy the descent condition as the current direction to explore for the minimization objective is a linear combination of the gradient vector and the previous vector i.e,

$$d_k = \begin{cases} -g_k, & k = 0 \\ -g_k + \beta_k d_{k-1}, & k \geq 1 \end{cases} \tag{3.9}$$

where $g_k = \nabla f(x_k)$ and β_k is known as the CG coefficient. There are many ways to calculate β_i and some well-known formulae are:

$$\beta_k^{FR} = \frac{g_k^T g_k}{||g_{k-1}||^2}$$

$$\beta_k^{PR} = \frac{g_k^T (g_k - g_{k-1})}{||g_{k-1}||^2}$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}}$$

where g_k and g_{k-1} are gradients of $f(x)$ at the points x_k and x_{k-1}, respectively, while $||.||$ is a norm of vectors and d_{k-1} is a direction for the previous iteration. The above

corresponding coefficients are known as Fletcher and Reeves (23), Polak and Ribiere (24), Hestenes and Stiefel (25), Bamigbola-Ali-Nwaeze (26), Dai and Yuan (27)
The algorithm (3.1) for conjugate gradient method is as below.
Algorithm 2.1: The algorithm for conjugate gradient method
Step 1. Start with an arbitrary initial point x_0
Step 2. Set the initial search direction $d_0 = -g_0$
Step 3. Find the point x_1 according to the relation $x_1 = x_0 + \alpha_0 d_0$
where α_0 is the optimal step length in the direction d_0 set $k = 1$and go to the next step.
Step 4. Find $g_k = g(x_k)$ and set $d_k = -g_k + \beta_k d_{k-1}$.
Compute the optimum step length α_k in the direction d_k and find the new point $x_{k+1} = x_k + \alpha_k d_k$.

At the current iterate point x_k determinate a search direction

$$d_k = \begin{cases} -(\bar{Q}x_k + \bar{c}), & for \quad k = 0 \\ -(\bar{Q}x_k + \bar{c}) + \beta_k d_{k-1} & for k \geq 1 \end{cases} \tag{3.10}$$

where β_k is chosen such that d_k is a conjugate direction ofd_{k-1} with respect to the matrix $\bar{Q}$.
Along the direction d_k choose a step size $\alpha_k$such that, at the new iterate point

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.11}$$

the absolute value of the function $F(x;\theta)$decreases sufficiently.
The following lemma presents a method to determine the search direction.
Lemma 3.1. If

$$\beta_k = \frac{(\bar{Q}x_k + \bar{c})\bar{Q}d_{k-1}}{d_{k-1}\bar{Q}d_{k-1}} \tag{3.12}$$

$$d_k = -(\bar{Q}x_k + \bar{c}) + \beta_k d_{k-1} \tag{3.13}$$

then $d_k$in (20)is a conjugate direction of $d_{k-1}$with respect to $\bar{Q}$
Proof:Owing to

$$\begin{aligned} (d_k)^T \bar{Q} d_{k-1} &= -((\bar{Q}x_k + \bar{c}) + \beta_k d_{k-1})^T \bar{Q} d_k - 1 & (3.14) \\ &= (-\bar{Q}x_k + \bar{c}) + (\frac{(\bar{Q}x_k + \bar{c})^T \bar{Q} d_{k-1} d_{k-1}}{(d_{k-1})^T \bar{Q} d_{k-1}})^T \bar{Q} d_{k-1} & (3.15) \\ & & (3.16) \end{aligned}$$

the desired result is obtained. Actually, the formula (3.12) is called HS method. In the case that the step size α_k is chosen by exact linear search along the direction d_k, that is,

$$\alpha_k = -\frac{(\bar{Q}x_k + \bar{c})^T d_k}{(d_{k-1})^T \bar{Q} d_k} \tag{3.17}$$

$$d_k = \begin{cases} -(\bar{Q}x_k + \bar{c}), & for \quad k = 0 \\ -(\bar{Q}x_k + \bar{c}) + \beta_k^{FR} d_{k-1} - \beta_k^{FR} (\frac{(\bar{Q}x_k + \bar{c})^T d_{k-1}}{(\bar{Q}x_k + \bar{c})^T) \bar{Q} x_k + \bar{c}} \bar{Q}x_k + \bar{c}) & for \quad k \geq 1 \end{cases} \tag{3.18}$$

where $\beta_k^{FR} = \frac{(\bar{Q}x_k + \bar{c})^T \bar{Q} x_k + \bar{c}}{||\bar{Q}x_{k-1} + \bar{c}||^2}$

Algorithm 3.2 Penalty Based on Three-term Conjugate Gradient method.
Step 0.(Initialization) Given a starting point $\theta > 1$,$\lambda \in [0, 1]$, $\sigma > 0$, $\epsilon > 0$ and ρ. Input the expected return vector μ , and compute Q and c. Choose an initial solution x_0. Set $k = 0$, $i = 0$, and $x_k = x_i$.

Step 1 (Reformulation).If

$$||\bar{Q}x_k + \bar{c}|| \leq \epsilon \tag{3.19}$$

then set

$$x_i = x_k \tag{3.20}$$

and go to step4; otherwise, go to step2
Step 2(Search Direction). Compute the search direction d_k by (3.18).
Step3(inexact line Search). Compute α_k by (3.8) and update

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.21}$$

Return to step 1.
Step 4 (Feasibility Test). Check feasibility of x_i in problem(2.2). If

$$P(x_i; \theta) \leq \sigma$$

the algorithm terminates; otherwise, go to step 5.
Step 5 (Update). Set $i = 0$, $x_i = x_k$, $\theta = \rho\theta$. At the new iterate point x_i, modify the matrix Q and the vector c by (3.2) and (3.4),respectively. Set $i = i+1$, and return to step 1
Remark 3.4(1)In Algorithm 1, the index i denotes the number of updating penalty parameter, and k denotes the number of iterations of three-term conjugate gradient method for unconstrained subproblem(3.6)
(2) For some fixed θ, it is easy to see that the condition

Hence, we need to make a few assumption based on the objective function
Assumption 3.1
$\bar{Q}x_1$: The biobjective function f is twice continuously differentiable.
$\bar{Q}x_2$: The level set L is convex. Moreover, positive constants c_1 and c_2 exist, satisfying

$$c_1||z||^2 \leq z^T F(x) z \leq c_2||z||^2 \tag{3.22}$$

for all $z \in R^n$ and $x \in L$ where $f(x)$ is the Hessian matrix of f.
$\bar{Q}x_3$: The Hessian matrix is Lipschitz continuous at the point x^* that is, there exist the positive constant c_3 satisfying

$$||g(x) - g(x^*)|| \leq c_3||x - x^*|| \tag{3.23}$$

for all x in a neighborhood of x^*

$$P(x_i; \theta_i) = \frac{\theta_i}{2}[(e^T x_i - 1)^2 + ||min(x_i - a, 0)||^2 + ||min(b - x_i, 0)||^2 \leq \sigma \tag{3.24}$$

implies that x_i is feasible. From theorem(2.3), it leads that x_i is a global minimizer of the original problem(1.5). If x_i is a global minimizer of problem(2.3)
Lemma 3.1
In the CG method,

$$g_k^T d_{k-1} = 0. \tag{3.25}$$

where g_k denotes the corresponding gradient.

The following definitions are prerequisites to the proceeding analysis.
Definitions
The search direction d_k is said to satisfy
(i) the descent condition if

$$g_k^T d_k < 0 \tag{3.26}$$

(ii) the sufficient descent condition if there exists a constant $c > 0$ such that

$$g_k^T d_k \leq -c||g_k||^2 \tag{3.27}$$

Lemma 3.2
The three-term CG family is a set of descent methods.
Proof
let $g_k = \bar{Q}x_k + \bar{c}$

$$(d_k)^T \bar{Q} d_{k-1} = -(g_k) + \beta_k d_{k-1} - \beta_k (\frac{(g_k)^T d_{k-1}}{(g_k)^T g_k} g_k)^T \bar{Q} d_{k-1} \tag{3.28}$$

$$(d_k)^T \bar{Q} d_{k-1} = -(g_k) + \beta_k (d_{k-1} - (\frac{(g_k)^T d_{k-1}}{(g_k)^T) g_k} g_k)^T \bar{Q} d_{k-1} \tag{3.29}$$

$$(d_k)^T \bar{Q} d_{k-1} = 0 \tag{3.30}$$

Lemma 3.3
Proof

$$g_k^T d_k = -g_k^T (g_k) + \beta_k d_{k-1} - \beta_k (\frac{(g_k)^T d_{k-1}}{(g_k)^T) g_k} g_k) \tag{3.31}$$

$$g_k^T d_k = -g_k^T g_k \quad \text{by Lemma 3.1} \tag{3.32}$$

$$g_k^T d_k = -c\|g_k\|^2 \tag{3.33}$$

$$g_k^T d_k < 0 \quad since \quad c > 0 \tag{3.34}$$

Theorem 3.3 (Global convergence)
Lemma 3.3
The optimal search parameter $c^* = \frac{3}{4}$.
The sufficient descent condition is proved as
Proof
Lemma 3. Consider any method 3.18, where $\beta_i = \beta_i^{FR}$. we get

$$g_i^T d_i \leq -\frac{3}{4}||g_i||^2 \tag{3.35}$$

Proof.

$$d_k = (\bar{Q}x_k + \bar{c}) + \beta_k^{FR} d_{k-1} - \beta_k^{FR} (\frac{(\bar{Q}x_k + \bar{c})^T d_{k-1}}{(\bar{Q}x_k + \bar{c})^T) \bar{Q}x_k + \bar{c}} \bar{Q}x_k + \bar{c}) \tag{3.36}$$

$$\beta_k^{FR} = \frac{(\bar{Q}x_k + \bar{c})^T \bar{Q}x_k + \bar{c}}{||\bar{Q}x_{k-1} + \bar{c}||^2} \tag{3.37}$$

put (3.25) in (3.23) and multiply through by g_k^T
let $g_k^T = (\bar{Q}x_k + \bar{c})^T$

$$g_k^T d_k = -g_k^T (g_k) + \frac{g_k^T g_k}{||g_{k-1}||^2} d_{k-1} g_k^T - \frac{(g_k^T g_k)}{||g_{k-1}||^2} g_k^T (\frac{(g_k^T d_{k-1})}{g_k^T g_k} g_k) \tag{3.38}$$

From the conjugacy property of CGM, $g_k^T d_{k-1} = 0$. Then(3.27) becomes

$$g_k^T d_k = -g_k^T(g_k) + \frac{g_k^T g_k}{||g_{k-1}||^2} g_k^T d_{k-1} - \frac{(g_k^T g_k)}{||g_{k-1}||^2} g_k^T (\frac{(g_k^T d_{k-1})}{g_k^T g_k} g_k) \tag{3.39}$$

We apply the inequality $U^T V \leq \frac{1}{2}(||U||^2 + ||V||^2)$ to get

$$g_k^T d_k = -g_k^T(g_k) + \frac{g_k^T g_k}{||g_{k-1}||^2} d_{k-1} g_k^T \tag{3.40}$$

$$\frac{g_k^T g_k}{||g_{k-1}||^2} d_{k-1} g_k^T = \frac{\sqrt{2}}{\sqrt{2}} \frac{g_k^T g_k}{||g_{k-1}||^2} d_{k-1} g_k^T \tag{3.41}$$

Let
$U = \frac{1}{\sqrt{2}} g_k^T d_{k-1}^T g_k d_{k-1}$
$V = \sqrt{2} g_k^T d_{k-1} g_k$
From the conjugacy property of CGM, $g_k^T d_{k-1} = 0$. Then(3.27) becomes
$U^T V \leq \frac{1}{2(d_{k-1}^T g_k)^2}(\frac{1}{2}(d_k^T g_k)^2 ||g_k||^2 + 2((g_k^T d_{k-1})^2 ||g_k||^2)$

$$\frac{g_k^T g_k}{||g_{k-1}||^2} d_{k-1} g_k^T \leq \frac{1}{4}||g_k||^2 \tag{3.42}$$

Putting (3.40) in (3.38) we have

$$g_k^T d_k \leq -||g_k||^2 + \frac{1}{4}||g_k||^2 \tag{3.43}$$

Hence

$$g_k^T d_k \leq -\frac{3}{4}||g_k||^2 \tag{3.44}$$

Lemma (3.2). Under Assumption 3.1, positive constants ω_1 and ω_2 exist, such that for any x_k and d_k with $g_k^T d_k < 0$, the step size a_k produced by Algorithm (3.1) or (3.2) will satisfy either

$$f(x_k + \alpha_k d_k) - f_k \leq -\omega_1 \frac{(g_k^T d_k)^2}{||d_k||^2} \tag{3.45}$$

or

$$f(x_k + \alpha_k d_k) - f_k \leq \omega_1 g_k^T d_k \tag{3.46}$$

Proof. Suppose that $a_k < 1$, which means that (3.8) failed for a step size
$a' \leq a_k/\tau$:

$$f(x_k + \alpha_k' d_k) - f(x)_k \leq \omega a' g_k^T d_k \tag{3.47}$$

Then, by using the mean value theorem, we obtain

$$f(x_{k+1}) - f(x_k) = g^T(x_{k+1} - x_k) \tag{3.48}$$

where $g = \nabla f(x)$, for some $x \in (x_k, x_{k+1})$. Now by the Cauchy-Schwartz inequality, we get

$$g^T(x_{k+1} - x_k) = g^T(x_{k+1} - x_k) + (g - g_k)^T(x_{k+1} - x_k) \tag{3.49}$$

$$\leq g^T(x_{k+1} - x_k)g - g_k||(x_{k+1} - x_k) \tag{3.50}$$

$$\leq g^T(x_{k+1} - x_k) + \mu||x_{k+1} - x_k||^2 \tag{3.51}$$

$$\leq g^T(a' d_k) + \mu||a' d||^2 \tag{3.52}$$

$$\leq g^T(a' d_k) + \mu a'||d||)^2 \tag{3.53}$$

Thus, from H3

$$(\omega - 1)a^{'} g_k^T d_k < a^{'} (g - g_k)^T d_k \leq M(a^{'} ||d_k||)^2 \tag{3.54}$$

which implies that

$$a_k \geq \tau a^{'} > \tau(1 - \omega) \frac{-g_i^T d_k}{M(a^{'}||d_k||)^{2n}} \tag{3.55}$$

$$f(x_k + \alpha_k^{'} d_k) - f(x)_k \leq c_2 \frac{-g_k^T d_k}{a^T ||d_k||^2} \tag{3.56}$$

where $c_2 = \tau(1 - \omega)/M$, which gives (3.43)

4 Numerical Consideration

In this section we use a set of selected unconstrained optimization problems from the CUTEr suite. The results obtained using Penalty Algorithm Based on Three-Term Conjugate Gradient Method compared with Penalty Algorithm Based on Two-Term Conjugate Gradient Method are shown in figure 1 and 2. In our numerical experiments, the initial solution is chosen to satisfy.

$$e^T x^0 = 1 \tag{4.1}$$

the bound vector a is a vector of all zeros, and b is a vector of all ones. We take the initial penalty parameter $\theta = 10$ and the aversion coefficient $\lambda = 0.5$. The tolerance of error is taken as$\epsilon = 10^{-7}$
Each of the test problems is tested with dimensions varying from 2 to 1000. For the Armijo line search, we use $\sigma = 10^{-4}$, the stopping criteria used are $||g_i|| \leq 10^{-6}$ and the number of iterations exceeds a limit of 10,000. Performance profile were drawn for the above methods. In general $p(\tau)$ is the fraction of problems with performance ratio τ; thus, a solver with high values of $p(\tau)$ is preferable. The implementation, numerical tests was performed on Compaq Presario CQ57-339WM Notebook PC, Windows 7 operating system, and Matlab 2013 languages.

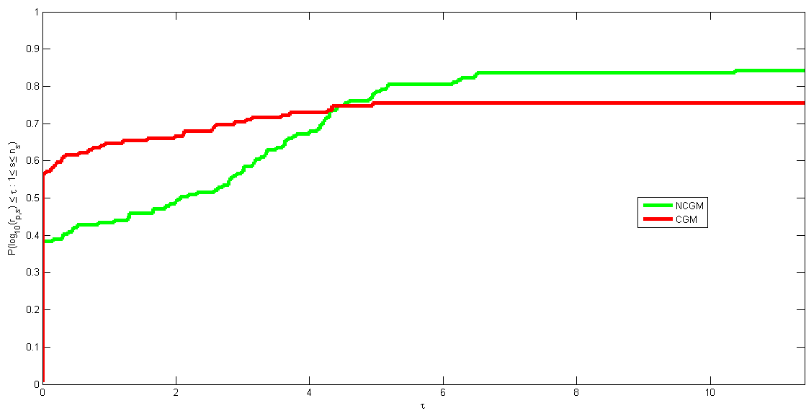

Figure 1: Performance Profile in a log_{10} scale based on iteration

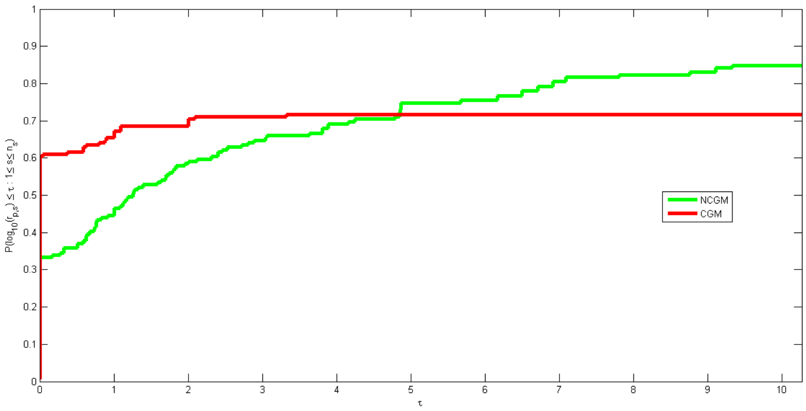

Figure 2:

Performance Profile in a log_{10} scale based on CPU time

Table 1: A list of problem functions

Test Problems	n-dimension	Sources
Powell badly scaled	2	More et al.[29]
Diagonal1 Function	2	More et al.[29]
Hager Function	6, 6	More et al.[29]
quartc FH1 Function	4, 6	More et al. [29]
Extended Wood Function	4	Michalewicz[30]
FLETCHCR Function	2, 4	More et al.[29]
SINQUAD Function	2	More et al.[29]
Power Functionl	2	Michalewicz [30]
Himmelblau	2	Andrei[28]
Extended Matyas Function	1, 2, 4 ,10 100	More et al.[29]
Extended Powell singular	4, 8	More et al. [29]
Extended Rosenbrock	2, 10, 100, 200, 500, 1000	Andrei[28]
Extended Hebert	2 4 10	Andrei[28]
Extended Cliff	2, 4, 10	More et al.[29]
Six-hump camel back polynomial	2	Michalewicz[30]
Extended Quadratic Penalty QP1	2, 4, 10, 100, 200, 500, 1000	Andrei [28]
Raydan 1	2, 4,	Andrei[28]
Raydan 2	2, 4 10 100 200	Andrei[28]
Extended Dixon and Price Function	2	Andrei[28]
Diagonal 9	2 4 10	Andrei[28]
PS1	2	Andrei[28]
Cube	2, 10, 100, 200	More et al.[29]

4.1 Remarks On Computational Results

Performance profiles of methods are illustrated in Figures 1 and 2. The performance profile seeks to find how well the solvers perform relative to the other solvers on a set of problems.

From Figure 1 and 2, we have that the NCGM method has the best performance since it can solve (80%) of the test problems compared with CGM

The computational results above shows that global convergence is achieved from different starting points on selected unconstrained optimization problems.

5 Final Remark

In this paper, the biobjectives optimization model of portfolio management was reformulated as an unconstrained minimization problem.Regarding the features of the optimization models in portfolio management, a class of penalty algorithms based on three-term conjugate gradient method was developed. The numerical performance of the proposed algorithm in solving the real problems verifies its effectiveness.

References

[1] H. Markowitz, Portfolio selection, *Journal of Finance*, vol. 7, pp. 7791, 1952.

[2] M. J. Best and H. Jaroslava, The efficient frontier for bounded assets, *Mathematical Methods of Operations Research*, vol. 52, no. 2, pp. 195212, 2000

[3] A. Yoshimoto, The mean-variance approach to portfolio optimization subject to transaction costs*Journal of the Operations Research Society of Japan*, vol. 39, no. 1, pp. 99117, 1996

[4] A. F. Perold, Large-scale portfolio optimization, *Management Science*, vol. 30, no. 10, pp. 11431160,1984.

[5] W. F. Sharpe, Portfolio Theory and Capital Markets, *McGraw-Hil, New York, NY, USA*, 1970

[6] E. D. Andersen, J. Gondzio, C. Meszaros, and X. Xu, Implementation of interior-point methods for large scale linear programs, *in Interior Point Methods of Mathematical Programming*, vol. 5 of Applied Optimization, pp. 189252, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[7] F. Potra, C. Roos, and T. Terlaky, Eds., Special Issue on Interior-Point Methods, *Optimization Method and Software*, vol. 11-12, 1999.

[8] T. F. Coleman and L. A. Hulbert, A direct active set algorithm for large sparse quadratic program swith simple bounds, *Mathematical Programming*, vol. 45, no. 3, pp. 373406, 1989.

[9] K. K. Lai, S. Y.Wang, J. P. Xu, S. S. Zhu, and Y. Fang, A class of linear interval programming problems and its application to portfolio selection,*IEEE Transactions on Fuzzy Systems, vol. 10, no. 6, pp. 698704, 2002.*

[10] Byrd R. H. and Nocedal J.:. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization, *SIAM Journal on Numerical Analysis 26(3), 727-739 (1989)*

[11] David H.: Applied nonlinear programming, *Mc Graw Hill, New York, 190-219.* (1972)

[12] Goldstein A. A.: On steepest descent, *SIAM Journal on Numerical Analysis 3*, 147-151 (1965)

[13] Ibrahim M. A. H., Mamat M. and Leong W.J.: The Hybrid BFGS-CG method in solving unconstrained optimization problems, *Hindawi Publishing Corporation, Article ID 507102, 6* (2014)

[14] Kamat M. P. and Hayduk R.J.: Recent developments in quasi-Newton methods for structural analysis and Synthesis, *American Institute of Aeronautics and Astronautics, 20*, 672-679 (1982)

[15] Ludwig A.:. The Gauss-Seidel-quasi-Newton method: a hybrid algorithm for solving dynamic economic model, *Journal of Economic Dynamics and Control 31*, 1610-1632 (2007)

[16] Rao S. S.: *Engineering optimization: Theory and practice, fourth edition*, John Wiley, New York (2009)

[17] Walsh G. R.: *Methods of optimization*, John Wiley, New York (1975)

[18] William W. H. and Hongchao Z.: A new conjugate gradient method with guaranteed descent and efficient line search, *SIAM Journal on Numerical Analysis 16*, 170-192 (2005)

[19] Gilbert J. C. and Nocedal J.:. Global convergence properties of conjugate gradient methods for optimization, *SIAM Journal on Numerical Analysis 2*, 21-42 (1992)

[20] J. S. Pang, A new and efficient algorithm for a class of portfolio selection problems, *Operations Research*, vol. 28, no. 3, part 2, pp. 754767, 1980.

[21] N. Kawadai and H. Konno, Solving large scale mean-variance models with dense non-factorable covariance matrices, *Journal of the Operations Research Society of Japan*, vol. 44, no. 3, pp. 251260, 2001.

[22] Z.J. Shi, Convergence of quasi-Newton method with new inexact line search, *J.Math. Anal. Appl*. 315, 120-131 (2006).

[23] Fletcher R. and Reeves R.M.: Function minimization by conjugate gradients, *The Computer Journal 7(2)*, 149-154 (1964)

[24] Polak E. and Ribiere G.: Note on the convergence of methods of conjugate direction, *Revue Francaised Informatique et de Recherche Operationnelle 3*, 35-43 (1969)

[25] Hestenes M. R. and Stiefel E.:. Method of conjugate gradient for solving linear equation, *Journal of Research of the National Bureau of Standards 49(6)*, 409-436 (1952)

[26] Bamigbola O. M., Ali M.M. and Nwaeze E.: A high-order nonlinear conjugate gradient method for unconstrained optimization, *Proceedings of International Congress of Mathematics*, Hyderabad, India (2010)

[27] Dai Y. H and Yuan Y.:. An efficient hybrid conjugate gradient method for unconstrained optimization, *Annals Operation Research. 103*, 33-47 (2001)

[28] Andrei N.: An unconstrained optimization test functions collection, *Advanced Modeling and Optimization 10(1), 147-161 2008*

[29] More J. J. , Garbow B. S. and Hillstrom K. E.: Testing unconstrained optimization software, *ACM Transaction on Mathematical Software 7*, 17-41 (1981)

[30] Michalewicz Z.: *Genetic Algorithms+Data Structures=Evolution programs*, Springer, Berlin (1996)