Spectral proximal method for solving large scale sparse optimization

Gillian Yi Han Woo[1*], *Hong Seng* Sim[1], *Yong Kheng* Goh[1], and *Wah June* Leong[2]

[1]Mathematical and Actuarial Sciences Department, Lee Kong Chian Faculty of Engineering and Science, University Tunku Abdul Rahman, 43000 Selangor, Malaysia
[2]Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Malaysia

Abstract. In this paper, we propose to use spectral proximal method to solve sparse optimization problems. Sparse optimization refers to an optimization problem involving the l_0-norm in objective or constraints. The previous research showed that the spectral gradient method is outperformed the other standard unconstrained optimization methods. This is due to spectral gradient method replaced the full rank matrix by a diagonal matrix and the memory decreased from $O(n^2)$ to $O(n)$. Since l_0-norm term is nonconvex and non-smooth, it cannot be solved by standard optimization algorithm. We will solve the l_0-norm problem with an underdetermined system as its constraint will be considered. Using Lagrange method, this problem is transformed into an unconstrained optimization problem. A new method called spectral proximal method is proposed, which is a combination of proximal method and spectral gradient method. The spectral proximal method is then applied to the l_0-norm unconstrained optimization problem. The programming code will be written in Python to compare the efficiency of the proposed method with some existing methods. The benchmarks of the comparison are based on number of iterations, number of functions call and the computational time. Theoretically, the proposed method requires less storage and less computational time.

1 Introduction

There has been an increased interest in the general field of sparsity in the past few years, [1, 2, 3]. Solving the sparse optimization to underdetermined linear systems has become a popular research topic in the area of compressive sensing, image processing, machine learning and statistics [4, 5]. Sparse optimization is an optimization problem which involves the zero-norm in objective or constraints. The zero-norm on R^n is denoted by l_0-norm or $\|.\|_0$, which represents the number of nonzero elements in $\mathbf{x}$. The l_0-norm plays an important and crucial role for modelling the sparsity of data and selecting representative variables in optimization problems. Mathematically, some people disagree l_0-norm as a proper norm because it does not satisfy the property of a norm. For all $\mathbf{x} \in R^n$ and $\lambda \neq 0$, one has $\|\lambda\mathbf{x}\|_0 = \|\mathbf{x}\|_0$, which is not being absolutely homogeneous shows that it is not a norm [5]. However, in this paper, we will adopt it as a norm [6].

*Corresponding author: gillianwoo@1utar.my

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

Sparsest solution can be found by applying sparse optimization which involves minimizing of the l_0-norm.

$$\|x\|_0 + \varphi(\mathbf{x}), \tag{1}$$

where the loss function, $\varphi(\mathbf{x})$ is the data fidelity term which relates to the problems in signal and image processing field [7]. For examples, the least-absolute (LA) loss function, $\|A(\mathbf{x}) - \mathbf{b}\|_1$and the least square (LS) loss function, $\|A(\mathbf{x}) - \mathbf{b}\|_2^2$. Due to the nonconvexity and discontinuity of the l_0-norm, the resulting optimization problem is known to be intractable (see, e.g., [8] for the NP-hardness of the l_0-minimization over a linear system). The l_1-norm convex regularization is a common approach which replaces $\|\mathbf{x}\|_0$ with the $\|x\|_1 = \sum_{i=1}^{n}|x_i|$, as l_1-norm which is a convex relaxation of l_0-norm [9], especially in compressed sensing. The popularity of the l_1-relaxation is enormous due to the exact recovery property under some conditions [10]. This l_1-norm model has been popularly used in many areas of application. For instance, in the fields of radar systems [11], communications [12], computed tomography (CT) [13] and also magnetic resonant imaging (MRI) [14].

l_1-regularizer can be a loose relaxation of the l_0-norm and does not always provide the true relevant variables [15]. In l_0-norm, all nonzero entries have equal contributions. However, l_1-regularizer penalizes the amplitude uniformly and thus, it has a drawback of always underestimates high-amplitude components of x. Due to this, reconstruction failures might occur with the least measurements [15, 16] and leads to undesirable blocky images [17]. In term of sparsity, l_1-norm and l_0-norm models do not provide a similar performance. Hence, many research in the field of compressive sensing and low-rank matrix recovery suggested that better performances can be achieved by taking better approximations of the l_0-norm and matrix rank [7]. In many applications, the non-convex l_0-norm based regularization has more advantages than the convex l_1-norm, such as in the fields of image restoration [18], bioluminescence [19], CT [17], and MRI reconstruction [20].

In this paper, we propose to solve large scale sparse optimization problems from an optimization angle. The l_0-norm and its constraint are minimized directly with the proposed method. The paper is organised as follows: Section 2 reviews some standard optimization methods for solving unconstrained smooth optimization problems and also proximal method for solving non-smooth problems, followed by Section 3, spectral proximal method is proposed to solve for NP-hardness sparse optimization problems, which combines spectral gradient method and proximal method. In Section 4, we provide and discuss some results of the comparisons between spectral proximal method and the existing methods. An executable programming code has been developed to test the efficiency of the method. A short conclusion for this paper will be made in Section 5.

2 Gradient-based optimization methods

2.1 Standard optimization methods

The unconstrained optimization problem is as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}), \tag{2}$$

where f is a twice continuously differentiable function and gradient is denoted by $\boldsymbol{g}$.

In this paper, we are interested in solving the large-scale optimization problems which the Hessian of f is either not available or requires a large amount of storage. Gradient methods are usually used to solve large-scale unconstrained convex optimization problem (2) and involve iterative minimization technique for finding the minimizer, $\mathbf{x_{k+1}}$ by the following rule:

$$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}, \tag{3}$$

where $\mathbf{d_k}$ is the search direction and $\alpha_k > 0$ is the steplength.

The classical steepest descent method, also known as the gradient descent method, was proposed by Cauchy (1847) [21]. The steepest descent method is the simplest gradient method for solving large scale unconstrained optimization. This method moves in a zig-zag like path along the negative direction of gradient towards the local minimum point [22].

Steepest descent algorithm can be implemented easily and only a low storage, *O(n)* is required [22]. However, this method is relatively slow when close to the minimum. This is because near the local minimization the gradient is nearly zero, and thus the rate of descent is slow [23]. This method is important for the development of the theory of optimization but it is quite slow for most of the real-world problems. Hence, more powerful techniques such as conjugate gradient method and quasi-Newton methods are frequently used [24].

Conjugate gradient method is an iterative method for using conjugate directions having *Q*-orthogonality. Suppose f is a quadratic function $f(\mathbf{x}) = c + \mathbf{b^T x} + \frac{1}{2}\mathbf{x^T} Q\,\mathbf{x}$ with n variables, where Q is a symmetric positive definite matrix. *Q*-orthogonality states that the directions $\mathbf{d_0}, \mathbf{d_{1,\ldots}}, \mathbf{d_k}$ are a set of n mutually conjugate vectors with respect to Q, if $\mathbf{d_i^T Q d_j} = \mathbf{0}$ for all $\boldsymbol{i} \neq j$. The minimum of the function f will be found using these as directions of search in n or less iterations [25]. It surpasses the steepest descent as it avoids the repetitive steps and takes only Q-orthogonal steps. In general, conjugate gradient algorithm converges faster than basic steepest descent algorithms [26].

Quasi-Newton methods are the algorithms that are used to find the local maxima or minima of a function [27]. It is an improvement version of Newton's method to overcome its main drawback which are the difficulties in evaluating the inverse of the Hessian matrix. Since the inverse of the Hessian matrix is full rank matrix, the computational time and cost are higher. To avoid the complexity in computing the inverse of Hessian matrix, quasi-Newton methods use an approximation to replace an exact inverse Hessian matrix. There are two popular methods in quasi-Newton method: Broyden-Fletcher-Goldfarb-Shann (BFGS) and Davidon-Fletcher-Powell (DFP). In general, quasi-Newton method requires more storage and computational time compared to conjugate gradient method [28].

To overcome the disadvantages of these methods, multiple spectral gradient was proposed by Sim et al. [29] to minimize the usage of storage and computation time.

2.2 Spectral gradient method

DFP and BFGS were derived based on Frobenius norm,

$$\begin{aligned} &\min \frac{1}{2}\|H_{k+1} - H_k\|_2^2 \\ &\text{s.t. } \mathbf{s_k} = H_{k+1}\mathbf{y_k}, \end{aligned} \tag{4}$$

where $\mathbf{s_k} = \mathbf{x_{k+1}} - \mathbf{x_k}$ and $\mathbf{y_k} = \mathbf{g_{k+1}} - \mathbf{g_k}$. Notice that the constraint in (4) is the secant equation which is reversed from $B_{k+1}\mathbf{s_k} = \mathbf{y_k}$ and consists of the curvature condition [29].

The inverse of B_{k+1} which is denoted by

$$H_{k+1} = B_{k+1}^{-1}, \tag{5}$$

is the approximation of inverse Hessian matrix.

In spite of the DFP and BFGS algorithms avoid the need to compute the inverse of Hessian matrix for f, the methods still require $O(n^2)$ storage for the inverse of Hessian matrix. To overcome this, spectral gradient method was proposed by Sim et al. in 2019 [29]. This method requires only $O(n)$ storage by replacing full rank matrix, H_{k+1} with a diagonal matrix, B_{k+1}^{-1}. B_{k+1} is the approximation of the eigenvalues of the actual Hessian matrix.

To derive an updated scheme for B_k, a restriction for components of B_k under some measures is imposed by minimizing the log-determinant norm and allowing them to satisfy the secant equation in (4). Hence, for any positive matrix B, the solution is given by the updated B_{k+1}:

$$\min \operatorname{tr}(B_{k+1}) - \ln(\det(B_{k+1})) \tag{6}$$

$$\text{s.t. } \mathbf{s_k^T} B_{k+1} \mathbf{s_k} = \mathbf{s_k^T y_k}. \tag{7}$$

Let $B_{k+1} = diag\left(B_{k+1}^{(i)}, \dots, B_{k+1}^{(n)}\right)$ and $\mathbf{s_k} = \left(s_k^{(1)}, \dots, s_k^{(n)}\right)$. The minimization problem from (6) and (7) becomes

$$\min\left(\sum_{i=1}^{n} B_{k+1}^{(i)}\right) - \ln\left(\prod_{i=1}^{n}\left(B_{k+1}^{(i)}\right)\right) \tag{8}$$

$$s.t.\ \left(\sum_{i=1}^{n}\left(s_k^{(i)}\right)^2 B_{k+1}^{(i)}\right) - \mathbf{s_k^T y_k} = 0. \tag{9}$$

Using the Lagrange method, the Lagrangian function for (8) and (9) is formed:

$$L(\alpha, \omega) = \left(\sum_{i=1}^{n} B_{k+1}^{(i)}\right) - ln\left(\prod_{i=1}^{n}\left(B_{k+1}^{(i)}\right)\right) + \omega\left[\left(\sum_{i=1}^{n}\left(s_k^{(i)}\right)^2 B_{k+1}^{(i)}\right) - \mathbf{s_k^T y_k}\right], \tag{10}$$

where ω is the Lagrange multiplier. Differentiating (10) partially with respect to $B_{k+1}^{(i)}$ and equate the resulting to zero yields

$$B_{k+1}^{(i)} = \frac{1}{1+\omega\left(s_k^{(i)}\right)^2}, i = 1,2, \dots, n. \tag{11}$$

By substituting (11) into constraint (9) gives:

$$F(\omega) = \left(\sum_{i=1}^{n} \frac{\left(s_k^{(i)}\right)^2}{1+\omega\left(s_k^{(i)}\right)^2}\right) - \mathbf{s_k^T y_k}, \tag{12}$$

where the Lagrange multiplier ω can be obtained by applying only one iteration of Newton-Raphson, starting from $\underline{\omega} = 0$. When $\mathbf{s_k^T s_k} > \mathbf{s_k^T y_k}$, the equation (12) has a unique positive solution and hence, the Lagrange multiplier, ω_k can be approximated by:

$$\omega_k \approx \underline{\omega} - \frac{F(\underline{\omega})}{F'(\underline{\omega})}$$

$$= \frac{\mathbf{s_k^T s_k} - \mathbf{s_k^T y_k}}{\sum_{i=1}^{n}\left(s_k^{(i)}\right)^4}. \tag{13}$$

The updating formula for B_{k+1} is then given as follows:

$$B_{k+1} = \begin{cases} \text{diag}\left(B_{k+1}^{(1)}, \ldots, B_{k+1}^{(n)}\right), & if\ \mathbf{s_k^T s_k} > \mathbf{s_k^T y_k} \\ \frac{\mathbf{s_k^T y_k}}{\mathbf{s_k^T s_k}} I, & \text{otherwise,} \end{cases} \tag{14}$$

where $\frac{\mathbf{s_k^T y_k}}{\mathbf{s_k^T s_k}}$ is exactly the Oren-Luenberger scaling [30]. $B_{k+1}^{(i)}$ and ω are defined in (11) and (13) respectively.

The algorithm for spectral gradient method is showed as below:

Step 1. Set $k = 0$; given an initial guessing point $\mathbf{x_0}$, and a convergence tolerance ϵ. Compute $\mathbf{d_0} = -B_0^{-1}\mathbf{g_0}$, where B_0^{-1} is a $n \times n$ identity matrix.
Step 2. Set $k \geq 0$. If $\|\mathbf{g_k}\| \leq \epsilon$, stop, else compute α_k with backtracking line search strategy and $\mathbf{d_k} = -B_k^{-1}\mathbf{g_k}$ where B_k is defined by (14).
Step 3. Compute $\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}$.
Step 4. Set $k = k + 1$ and go to Step 2.

In Step 2, the backtracking line search (BTA) strategy with Armijo condition [31] is given as follows:

Step 1. Given constants $\eta \in (0,1)$ and α_1, α_2, with $0 < \tau_1 < \tau_2 < 1$.
Step 2. Initially, set $\alpha = 1$.
Step 3. Check whether the Armijo condition is satisfied:

$$f(\mathbf{x_k} + \alpha \mathbf{d_k}) \leq f(\mathbf{x_k}) + \eta \alpha \mathbf{g_k^T d_k} \tag{15}$$

Step 4. If (15) is not satisfied, choose a new $\alpha \in [\tau_1 \alpha, \tau_2 \alpha]$ and repeat Step 3. If (15) is satisfied, set $\alpha_k = \alpha$ and $\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{d_k}$.

2.3 Proximal method

Proximal method is not an optimization method. However, we will discuss it in this section as we borrow the proximal method for solving our problem. Here we review some basic ideas of proximal method.

The proximal method was introduced by Martinet [32] as a regularization method in the context of convex optimization in Hilbert spaces. Steepest descent, conjugate gradient and Newton's methods are standard tools for solving unconstrained smooth minimization problems of modest size, while proximal algorithms can be viewed as an analogous tool for solving non-smooth, constrained, large-scale, or distributed problems. They are especially well-suited to large or high-dimensional datasets problems. The base operations in classical methods are low-level which consist of linear algebra operations and the computation of gradients and Hessians while the base operation in proximal algorithms is evaluating the proximal operator of a function [33].

The proximal point mapping is the basis of many optimization techniques for convex functions. Recently, proximal algorithm was extended from convex optimization to non-convex optimization [34]. The non-convex and non-smooth function is a hard problem in optimization, it usually solves with tractable convex optimization based on a regularization or relaxation. Proximal method will be used to solve the non-convex problem directly.

3 Spectral proximal method

Sparse optimization refers to an optimization problem involving l_0-norm in objective or constraints. In this paper, we propose an efficient, general purpose algorithmic approach and efficient implementations for the following non-convex optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t. } A\mathbf{x} = \mathbf{b}, \tag{16}$$

where $A\mathbf{x} = \mathbf{b}$ is an underdetermined system. Since l_0 -norm is non-convex, non-differentiable and non-continuous, it cannot be solved by the standard tool which is applicable in solving the unconstrained smooth optimization. The constraint of sparse optimization is in the form $A\mathbf{x} = \mathbf{b}$, it is difficult to solve and requires large computation time when it has a high dimensional dataset.

The problem (16) usually arises in machine learning, neural networks, signal processing and bioinformatics. Hence, it motivated us to develop an efficient algorithm to solve for this problem.

Methodology of research is considered into two main parts. Firstly, the proximal method and spectral gradient method programming framework will be reviewed in order to examine how these methods can be extended to solve optimization problems involving l_0-norm and its constraints. The problem is transformed from constrained optimization problem into unconstrained optimization problem by Lagrange method:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 + \frac{\lambda}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2. \tag{17}$$

Instead of solving $A\mathbf{x} = \mathbf{b}$, we consider to minimize the residue of $A\mathbf{x} = \mathbf{b}$ which can be expressed in the form of $\min \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$ where l_2-norm regularization acts as a smoothing term for the residue $(A\mathbf{x} - \mathbf{b})$.

Secondly, spectral gradient method is modified and incorporated into the proximal method in solving the sparse optimization problems. Since l_0-norm is non-convex, non-differentiable and non-continuous, it will be solved by the proximal method. It is very generally applicable, but is especially suitable to problems involving large or high-dimensional datasets.

Proximal quasi newton has been proposed by Tang et al. previously [35]. On literature, there is no research has been done on spectral proximal method and proximal steepest descent method. To fill in the gap, we propose to solve the residue by applying spectral gradient method which is proposed by Sim et al. [29]. From the paper by Sim et al. [29], spectral gradient method outperformed than a list of conjugate gradient methods since spectral gradient method requires less storage and less computational time.

Spectral gradient method and proximal method are applied alternatively to solve the problem. The step length, α_k is obtained through Armijo condition as the line search strategy [31].

The algorithm is as follows:

Step 1. Initialize parameters.
Set $k = 0$; give an initial guess for x_0, set a convergence tolerance ϵ, and set the sparsity control parameter $\mu > 0$. Compute $\mathbf{d_0} = -B_0^{-1}\mathbf{g_0}$, where B_0^{-1} is an $n \times n$ identity matrix.

Step 2. Check stopping criteria.

Set $k \geq 0$. If $\|\mathbf{g_k}\| \leq \epsilon$, stop, else compute spectral gradient method to obtain $\mathbf{x_{k+1}}$.
Step 3. Check $\mathbf{x_{k+1}}$ by proximal method.

$$x_{k+1}^{(i)} = \begin{cases} 0 \ , & \text{if } x_{k+1}^{(i)} < \sqrt{2\mu} \\ x_{k+1}^{(i)}, & \text{if } x_{k+1}^{(i)} = \sqrt{2\mu} \\ x_{k+1}^{(i)}, & \text{if } x_{k+1}^{(i)} > \sqrt{2\mu} \end{cases}$$

Step 4. Set $k = k + 1$ and go to Step 2.

In short, the l_0-norm which is non-smooth is solved by proximal method and the residue which is smooth is solved by spectral gradient method. These two methods are combined to solve the problem and is named as spectral proximal method. Next, an executable code is developed using Python software to test the efficiency of the proposed method with the existing methods. The benchmarks of the comparison are based on number of iterations, number of functions call and the computational time.

4 Numerical experiments and discussion

In order to compare the efficiency of spectral proximal method with other methods, we developed an executable code using Python 3.7 software. The software is downloaded on Lenovo ideapad 330S, where all the experiments are executed with 8th Generation Intel Core i7 processor and 12 GB RAM. For all the experiments, we choose the initial point x_0 as a column matrix $(n \times 1)$ of ones. A is $m \times n$ matrix and B is $m \times 1$ matrix. For proximal method, μ is set as 10^{-8}. The maximum number of iterations is set at 5000. The method is considered failed when the number of iterations has exceeded 5000. Tolerance, ϵ is set to be 10^{-4} and algorithm is terminated when $\|\mathbf{g_k}\| \leq \epsilon$.

We generated 500 random matrices with dimensional $7 \times 10, 35 \times 50, 70 \times 100, 140 \times 200$ and 350×500, condition number is set from 1 until 20 with 5 different seeds for each of the dimension. The results from 5 different seeds with same dimensional and condition number's random matrices are averaged. The efficiency of the methods are based on number of iterations, number of functions call and the time consuming. The three methods are steepest descent with proximal (PSD), Broyden-Fletcher-Goldfarb-Shann with proximal (PBFGS) and spectral proximal method (PSG). The profiling graphs are shown below:

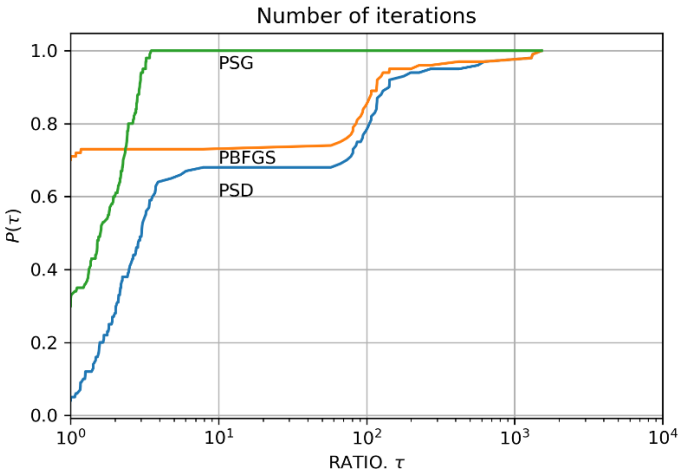

Fig. 1. Comparison of methods in terms of number of iterations.

Fig. 1 shows that PSD requires larger number of iterations when compares to the others. This is because it moves in zigzag form towards the local minimum point. It takes more steps and has slower convergence especially when it closes to the minimum point. PBFGS performs slightly better than PSD. Overall, PSG shows the best performance with lesser number of iterations.

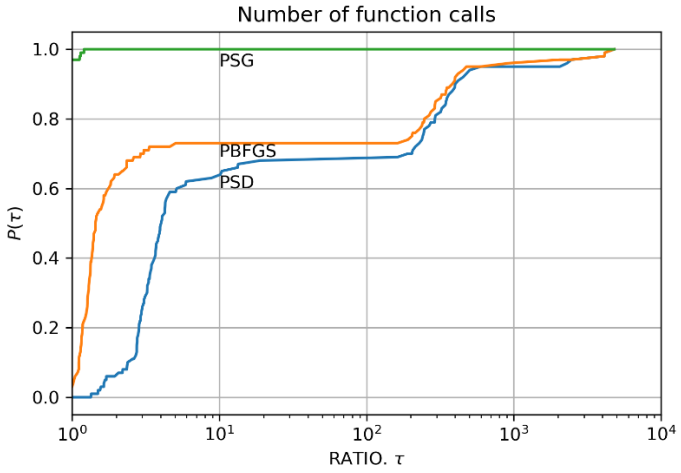

Fig. 2. Comparison of methods in terms of number of function calls.

Fig. 2 shows the profiling graph in terms of number of function calls. From the profiling graph above, we can conclude that the PSG method requires the least number of function calls to get the desired solution. On the other hand, the PBFGS method performs slightly better than PSD method. This is mainly due that PBFGS method and PSD method need more function calls in the backtracking line search strategy to obtain the optimum step size while PSG requires less function calls to get the similar results.

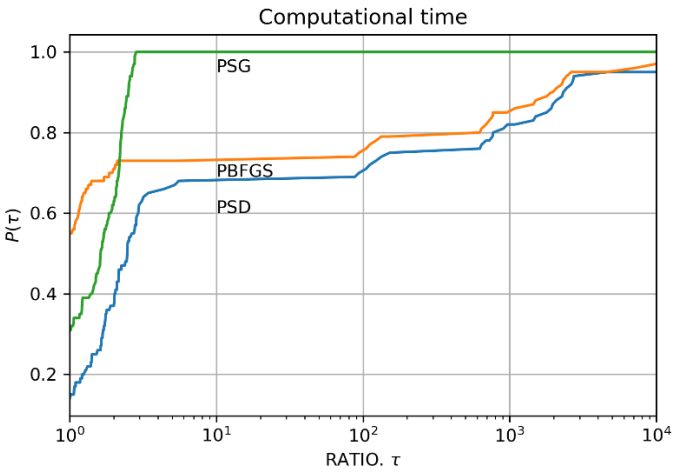

Fig. 3. Comparison of methods in terms of CPU time in seconds.

Fig. 3 shows the profiling graph in terms of computational time. PSD method requires more computational time because it moves in zigzag like path and thus has slower convergence. PSG requires less computational time compared to PBFGS due to the fact that in PBFGS, the Hessian matrix is approximated by a full rank matrix with storage memory $O(n^2)$; while in PSG, the eigenvalues of the Hessian matrix is approximated by a diagonal matrix with storage memory $O(n)$. Hence, PSG requires minimal storage and thus less computational time is needed.

5 Conclusions

A new method called spectral proximal method is proposed for solving large scale sparse optimization problem. Sparse optimization involves minimizing l_0-norm (a NP-hardness problem) which is non-convex and non-smooth. To solve this problem, we incorporated spectral gradient method with the proximal method. The numerical results show that the spectral gradient method requires less storage and less CPU time, hence this proposed method has a better performance than the other methods.

The research was supported by Ministry of Higher Education (MOHE) for financial supports through Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UTAR/02/4). We also want to thank to the Universiti Tunku Abdul Rahman which provide UTAR Research Fund (IPSR/RMC/UTARRF/2019-C1/S02) for sponsoring this work.

References

1. A. Sharma, S.S. Bhadauria, R. Gupta, *Image compression and sparsity measurement by using multilevel and different wavelet functions*, in 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 517 – 520 (2019)
2. Y. Li, S. Gu, C. Mayer, L.V. Gool, R. Timofte, *Group sparsity: the hinge between filter pruning and decomposition for network compression*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)

3. R. Blanquero, E. Carrizosa, C. Molero-Río, D.R. Morales, Eur. J. Oper. Res. **284**(1), 255 – 272 (2020)
4. W. Deng, W. Yin, Y. Zhang, *Group sparse optimization by alternating direction method*, Wavelets and Sparsity XV **8858**, 88580R (2013)
5. H.A. Le Thi, T. Pham Dinh, H.M. Le, X.T. Vo, Eur. J. Oper. Res. **244**(1), 26 – 46 (2015)
6. J. Xu, Y.B. Zhao, Optim. Methods. Softw. **35**(4), 1 – 19 (2020)
7. Y. Sun, X. Tan, X. Li, L. Lei, G. Kuang, Signal Processing **168**, 107369 (2020)
8. B.K. Natarajan, SIAM J. Comput. **24**(2), 227 – 234 (1995)
9. E.J. Candes. Comptes. Rendus. Math. **346**(9), 589 – 592 (2008)
10. E.J. Candes, T. Tao, IEEE Trans. Inform. Theory **51**(12), 4203 – 4215 (2005)
11. J. Yang, J. Thompson, X. Huang, T. Jin, Z. Zhou, IEEE Trans. Geosci. Remote Sensing **51**(2), 983 – 994 (2013)
12. C.R. Berger, Z. Wang, J. Huang, S. Zhou, IEEE Commun. Mag. **48**(11), 164 – 174 (2010)
13. Z. Chen, X. Jin, L. Li, G. Wang, Phys. Med. Biol. **58**(7), 2119 – 2141 (2013)
14. M. Lustig, J.M. Pauly, D. Donoho, Magn. Reason. Med. **58**(6), 1182 – 1195 (2007)
15. E.J. Candes, M.B. Wakin, S.P. Boyd, J. Fourier Anal. Appl. **14**(5–6), 877 – 905 (2008)
16. R. Chartrand, V. Staneva, Inverse Probl. **24**(3) (2008)
17. Y. Sun, J. Tao, Int. J. Imaging Syst. Technol. **24**(3), 215 – 223 (2014)
18. C. Bao, Z. Shen, B. Dong, L. Hou, X. Zhang, X. Zhang, Inverse Probl. **32**(11) (2016)
19. X. Zhang, Y. Lu, T. Chan, J. Sci. Comput. **50**(3), 519 – 535 (2012)
20. J. Trzasko, A. Manduca, IEEE Trans. Med. Imaging **28**(1),106 – 121 (2009)
21. A. Cauchy, Rend. Sci. **25,** 46 – 89 (1847)
22. X. Wang, IEEE Microw. Wirel. Compon. Lett. **12**, 24-26 (2008)
23. X.S. Yang, Nature-inspired optimization algorithms (Elsevier, 1 – 21, 2014)
24. R. Poisel, Electronic warfare target location methods (Artech House, 2012)
25. M.R. Hestenes, E.L. Stiefel, J. Res. Natl. Bur. Stand. **49**(6), 409 – 436 (1952)
26. S. El Hajji, N. Moukafih, G. Orhanou, Analysis of neural network training and cost functions impact on the accuracy of IDS and SIEM systems, International Conference on Codes, Cryptology, and Information Security, 433 – 451 (2019)
27. C.G. Broyden, Math. Comput. **21**(99), 368 – 381 (1967)
28. J. Nocedal, S.J. Wright, Numerical Optimization (Springer Science & Business Media, 2006)
29. H.S. Sim, W.J. Leong, C.Y. Chen, Optim. Lett. **13(3)**, 617 – 632 (2019)
30. D.G. Luenberger. Linear and nonlinear programming, 2nd ed. (Addison-Wesley, 1984)
31. L. Armijo, Pac. J. Math. **16**(1), 1 – 3 (1966)
32. B. Martinet, Rev. Fran¸caise Inf. Rech. Oper. **4**(R3), 154 – 158 (1970)
33. N. Parikh, S. Boyd, Found. Trends Optim. **1**(3), 127 – 239 (2014)
34. W. Hare, C. Sagastizábal, Math. Program **116**(1-2):221–258 (2009)
35. X. Tang, K. Scheinberg, *Efficient quasi-newton proximal method for large scale sparse optimization*, NIPS (2013)