Palestine Journal of Mathematics

Vol. 6(Special Issue: II) (2017) , 157–164

EFFICIENT HYBRID CONJUGATE GRADIENT METHOD FOR SOLVING SYMMETRIC NONLINEAR EQUATIONS

Jamilu Sabi'u

Communicated by Ayman Badawi

MSC 2010 Classifications: Primary 90C30,90C26.

Keywords and phrases: Nonmonotone line search, Jacobian matrix, symmetric nonlinear equations, Conjugate gradient method.

Abstract In this article, two prominent conjugate gradient (CG) parameters were hybridized to proposed an efficient solver for symmetric nonlinear equations without computing exact gradient and Jacobian with a very low memory requirement. The global convergence of the proposed method was also established under some mild conditions with nonmonotone line search. Numerical results show that the method is efficient for large-scale problems.

1 Introduction

Let us consider the systems of symmetric nonlinear equations

$$F(x) = 0, \tag{1.1}$$

where $F : R^n \to R^n$ is a nonlinear mapping. Often, the mapping, F is assumed to satisfying the following assumptions:
A1. There exists an $x^* \in R^n$ s.t $F(x^*) = 0$
A2. F is a continuously differentiable mapping in a neighborhood of x^*
A3. $F'(x^*)$ is invertible
A4. The Jacobian $F^{'}(x)$ is symmetric.
where the symmetry means that the Jacobian $J(x) := F^T(x)$ is symmetric; that is, $J(x) = J(x)^T$. This class of special equations come from many practical problems such as an unconstrained optimization problem, a saddle point problem, Karush-Kuhn-Tucker (KKT) of equality constrained optimization problem, the discritized two-point boundary value problem, the discritized elliptic boundary value problem, and etc. Equation (1.1) is the first-order necessary condition for the unconstrained optimization problem where F is the gradient mapping of some function $f : R^n \longrightarrow R$,

$$min f(x), \quad x \epsilon R^n. \tag{1.2}$$

A large number of efficient solvers for large-scale symmetric nonlinear equations have been proposed, analyzed, and tested by different researchers. Among them are [4, 2, 10]. Still the matrix storage and solving of n-linear system are required in the BFGS type methods presented in the literature. The recent designed nonmonotone spectral gradient algorithm [1] falls within the frame work of matrix-free.
The conjugate gradient methods for symmetric nonlinear equations has received a good attension and take an appropriate progress. However, Li and Wang [5] proposed a modified Flectcher-Reeves conjugate gradient method which is based on the work of Zhang et al. [3], and the results illustrate that their proposed conjugate gradient method is promising. In line with this development, further studies on conjugate gradient are [7, 8, 11, 9, 13]. Extensive numerical experiments showed that each over mentioned method performs quite well. Therefore, motivated by [7] this article is aim at developing a derivative-free conjugate gradient method for solving symmetric nonlinear equations without computing the Jacobian matrix with less number of iterations and CPU time.

this paper is organized as follows: Next section presents the details of the proposed method. Convergence results are presented in Section 3. Some numerical results are reported in Section

4. Finally, conclusions are made in Section 5.

2 Efficient Hybrid Conjugate Gradient Method

Recall that, in [13] we used the term

$$g_k = \frac{F(x_k + \alpha_k F_k) - F_k}{\alpha_k} \tag{2.1}$$

to approximate the gradient $\nabla f(x_k)$, which avoids computing exact gradient. Also recall that, the method in [7] generates the sequence $x_{k+1} = x_k + \alpha_k d_k$, where the search direction d_k is given by

$$d_k = \begin{cases} -\nabla f(x_k) & if \quad k = 0 \\ -\nabla f(x_k) + \beta_k^{PRP} d_{k-1} - \theta_k^{PRP} y_{k-1} & if \quad k \geq 1 \end{cases} \tag{2.2}$$

where g_k is defined by (2.1), $y_k = F(x_k + \gamma_k) - F_k$, $\gamma_k = F_k - F_{k-1}$ and

$$\beta_k = \beta_k^{PRP} = \frac{\nabla f(x_k)^T y_{k-1}}{\|\nabla f(x_{k-1})\|^2} \quad \theta_k^{PRP} = \frac{\nabla f(x_k)^T d_{k-1}}{\|\nabla f(x_{k-1})\|^2}, \tag{2.3}$$

$||.||$ is the Euclidean norm.
From now on, problem (1.1) is assume to be symmetric and $f(x)$ is defined by

$$f(x) = \frac{1}{2}||F(x)||^2. \tag{2.4}$$

Then the problem (1.1) is equivalent to the global optimization problem (1.2). However, when $f(x)$ is given by (2.4):

$$\nabla f(x_k) = J(x_k)^T F(x_k) = J(x_k) F(x_k) \tag{2.5}$$

which requires the computions of both the Jacobian and the gradient of f. Recall that, from [6], they defined $\beta_k^{HS} = \frac{\nabla f(x_k)^T y_{k-1}}{d_{k-1}^T y_{k-1}}$ and $\theta_k^{HS} = \frac{\nabla f(x_k)^T d_{k-1}}{d_{k-1}^T y_{k-1}}$, now we defined efficient hybrid direction as:

$$d_k = \begin{cases} -\nabla f(x_k) & if \quad k = 0, \\ -\nabla f(x_k) + \beta_k^{H*} d_{k-1} - \theta_k^{H*} y_{k-1} & if \quad k \geq 1, \end{cases} \tag{2.6}$$

where

$$\beta_k^{H*} = \frac{\nabla f(x_k)_k^T y_{k-1}}{\max\{d_{k-1}^T y_{k-1}, \|\nabla f(x_{k-1})\|^2\}}, \quad and \quad \theta_k^{H*} = \frac{\nabla f(x_k)^T d_{k-1}}{\max\{d_{k-1}^T y_{k-1}, \|\nabla f(x_{k-1})\|^2\}}. \tag{2.7}$$

Replacing the terms $\nabla f(x_k)$ in(2.6)and (2.7) by (2.1), therfore β_k^{H*} becomes

$$\beta_k^{H*} = \frac{g_k^T y_{k-1}}{\max\{d_{k-1}^T y_{k-1}, \|g_{k-1}\|^2\}}, \quad and \quad \theta_k^{H*} = \frac{g_k^T d_{k-1}}{\max\{d_{k-1}^T y_{k-1}, \|g_{k-1}\|^2\}}. \tag{2.8}$$

Moreover, the direction d_k given by (2.6) may not be a descent direction of (2.4), then the standard wolfe and Armijo line searches can not be used to compute the stepsize directly. Therefore, the nonmonotone line search used in [11, 12, 13] is the best choice to compute the stepsize α_k. Let $\omega_1 > 0$, $\omega_2 > 0$, $r \in (0, 1)$ be constants and $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \infty. \tag{2.9}$$

Let $\alpha_k = max\,\{1, r^k\}$ that satisfy

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\omega_1||\alpha_k F(x_k)||^2 - \omega_2||\alpha_k d_k||^2 + \eta_k f(x_k). \tag{2.10}$$

Algorithm 1
Step 1 : Given x_0, $\alpha_k > 0$, $\omega \in (0, 1)$, $r \in (0, 1)$ and a positive sequence η_k satisfying (2.9), then compute $d_0 = -g_0$ and set $k = 0$.
Step 2 **:** Test a stopping criterion. If yes, then stop; otherwise continue with Step 3.
Step 3 : Compute α_k by the line search (2.10).
Step 4 : Compute $x_{k+1} = x_k + \alpha_k d_k$.
Step 5 : Compute the search direction by (2.6).
Step 6 : Consider $k = k + 1$ and go to step 2.

3 Convergence Result

This section presents global convergence results of an efficient hybrid CG method. To begin with, defined the level set

$$\Omega = \{x|f(x) \leq e^{\eta} f(x_0)\}, \tag{3.1}$$

where η satisfies

$$\sum_{k=0}^{\infty} \eta_k \leq \eta < \infty \tag{3.2}$$

.

Lemma 3.1. *[4] Let the sequence $\{x_k\}$ be generated by algorithm 1. Then the sequence $\{||F_k||\}$ converges and $x_k \epsilon \Omega$ for all $k \geq 0$.*

Proof. For all k, from (2.10) we have $\|F_{k+1}\| \leq (1 + \eta_k)^{\frac{1}{2}} \|F_k\| \leq (1 + \eta_k)\|F_k\|$. Since η_k satisfies (2.9), we conclude that $\{\|F_k\|\|$ converges. Moreover, we have for all k

$$\begin{aligned}
\|F_{k+1}\| &\leq (1 + \eta_k)^{\frac{1}{2}} \|F_k\| \\
&\vdots \\
&\leq \prod_{i=0}^{k} (1 + \eta_k)^{\frac{1}{2}} \|F_0\| \\
&\leq \|F_0\| \left[\frac{1}{k+1} \sum_{i=0}^{k} (1 + \eta_i) \right]^{\frac{k+1}{2}} \\
&\leq \|F_0\| \left[1 + \frac{1}{k+1} \sum_{i=0}^{k} \eta_i \right]^{\frac{k+1}{2}} \\
&\leq \|F_0\| \left(1 + \frac{\eta}{k+1} \right)^{\frac{k+1}{2}} \leq \|F_0\| \left(1 + \frac{\eta}{k+1} \right)^{k+1} \\
&\leq e^{\eta} \|F_0\|,
\end{aligned}$$

where η is a constant satisfying (2.9). This implies that $x_k \in \Omega$.
In order to get the global convergence of DFCG algorithm, we need the following assumptions.
(i) The level set Ω defined by (3.1) is bounded
(ii) In some neighbourhood N of Ω, the Jacobian of F is symmetric, bounded and positive definite. Namely, there exists a constant $L > 0$ such that

$$||J(x) - J(y)|| \leq L||x - y||, \quad \forall x, y \in N. \tag{3.3}$$

Li and Fukushima in [4] showed that, there exists positive constants M_1, M_2 and L_1 such that

$$||F(x)|| \leq M_1, \quad ||J(x)|| \leq M_2, \quad \forall x \epsilon N, \tag{3.4}$$

$$||\nabla f(x) - \nabla f(y)|| \leq L_1||x - y||, \quad ||J(x)|| \leq M_2, \quad \forall x, y\epsilon N. \tag{3.5}$$

Lemma 3.2. *Let the properties of* (1.1) *above hold. Then we have*

$$\lim_{k\to\infty} ||\alpha_k d_k|| = \lim_{k\to\infty} ||s_k|| = 0, \tag{3.6}$$

and

$$\lim_{k\to\infty} ||\alpha_k F_k|| = 0. \tag{3.7}$$

Proof. by (2.9) and (2.10) we have for all $k > 0$,

$$\omega_1||\alpha_k F(x_k)||^2 + \omega_2||\alpha_k d_k||^2 \leq f(x_k) - f(x_{k+1}) + \eta_k f(x_k), \tag{3.8}$$

by summing the above k inequality, then we obtain:

$$\sum_{i=0}^{m} \omega_1||\alpha_k F(x_k)||^2 + \omega_2||\alpha_k d_k||^2 \leq f(x_1) - f(x_m) + \sum_{i=0}^{m} \eta_i f(x_k). \tag{3.9}$$

So, from (3.5) and the fact that $\{\eta_k\}$ satisfies (2.9) the result follows.
The following result shows that algorithm 1 is globally convergent.

Theorem 3.3. *Let the properties of* (1.1) *hold. Then the sequence* $\{x_k\}$ *be generated by algorithm 1 converges globally, that is,*

$$\liminf_{k\to\infty} ||\nabla f(x_k)|| = 0. \tag{3.10}$$

Proof. We prove this theorem by contradiction. Suppose that (3.10) is not true, then there exists a positive constant τ such that

$$||\nabla f(x_k)|| \geq \tau, \quad \forall k \geq 0. \tag{3.11}$$

Since $\nabla f(x_k) = J_k F_k$, (3.11) implies that there exists a positive constant τ_1 satisfying

$$||F_k|| \geq \tau_1, \quad \forall k \geq 0. \tag{3.12}$$

Case (i): $\limsup_{k\to\infty} \alpha_k > 0$. then by (3.6), we have $\liminf_{k\to\infty} ||F_k|| = 0$. This and Lemma (3.1) show that $\lim_{k\to\infty} ||F_k|| = 0$, which contradicts with (3.11).

Case (ii): $\limsup_{k\to\infty} \alpha_k = 0$. Since $\alpha_k \geq 0$,this case implies that

$$\lim_{k\to\infty} \alpha_k = 0. \tag{3.13}$$

by definition of g_k in (2.1) and the symmetry of the Jacobian, we have

$$\begin{aligned} ||g_k - \nabla f(x_k)|| &= ||\frac{F(x_k + \alpha_{k-1}F_k) - F_k}{\alpha_{k-1}} - J_k^T F_k|| \\ &= ||\int_0^1 J(x_k + t\alpha_{k-1}F_k) - J_k)dtF_k|| \\ &\leq LM_1^2\alpha_{k-1}, \end{aligned} \tag{3.14}$$

where we use (3.4) and (3.5) in the last inequality. (2.9), (2.10) and (3.11) show that there exists a constant $\tau_2 > 0$ such that

$$||g_k|| \geq \tau_2, \quad \forall k \geq 0. \tag{3.15}$$

By (2.1) and (3.4), we get

$$||g_k|| = \|\int_0^1 J(x_k + t\alpha_{k-1}F_k)F_k dt\| \leq M_1 M_2, \quad \forall k \geq 0. \tag{3.16}$$

From (3.16) and (3.5), we obtain

$$||y_k|| = ||g_k - g_{k-1}||$$
$$\leq ||g_k - \nabla f(x_k)|| + ||g_{k-1} - \nabla f(x_{k-1})|| + ||\nabla f(x_k) - \nabla f(x_{k-1})||$$
$$\leq LM_1^2(\alpha_{k-1} + \alpha_{k-2}) + L_1||s_{k-1}||. \tag{3.17}$$

This together with (3.13) and (3.6) shows that $\lim_{k\to\infty} ||y_k|| = 0$. Again from the definition of our β_k^{H*} we obtain

$$|\beta_k^{H*}| \leq \frac{\|g_k^T\|}{\max\{\|d_{k-1}^T\|\|y_{k-1}\|, \|g_{k-1}\|^2\}} \leq \frac{M_1 M_2}{\max\{LM_1^2(\alpha_{k-1} + \alpha_{k-2}) + L_1||s_{k-1}||, M_1 M_2\}}||y_{k-1}|| \longrightarrow 0 \tag{3.18}$$

which implies there exists a constant $\rho \in (0, 1)$ such that for sufficiently large k

$$|\beta_k^{H*}| \leq \rho. \tag{3.19}$$

Without lost of generality, we assume that the above inequalities holds for all $k \geq 0$. Clearly its not difficult to see that θ_k^{H*} is bounded, also from (3.19) and (3.17) we can conclude that the sequence $\{d_k\}$ is bounded. Since $\lim_{k\to\infty} \alpha_k = 0$, then $\alpha_k^{'} = \frac{\alpha_k}{r}$ does not satisfy (2.10), namely

$$f(x_k + \alpha_k^{'} d_k) > f(x_k) - \omega_1||\alpha_k^{'} F(x_k)||^2 - \omega_2||\alpha_k^{'} d_k||^2 + \eta_k f(x_k), \tag{3.20}$$

which implies that

$$\frac{f(x_k + \alpha_k^{'} d_k) - f(x_k)}{\alpha_k^{'}} > -\omega_1||\alpha_k^{'} F(x_k)||^2 - \omega_2||\alpha_k^{'} d_k||^2. \tag{3.21}$$

By the mean-value theorem, there exists $\delta_k \in (0, 1)$ such that

$$\frac{f(x_k + \alpha_k^{'} d_k) - f(x_k)}{\alpha_k^{'}} = \nabla f(x_k + \delta_k \alpha_k^{'} d_k)^T d_k. \tag{3.22}$$

Since $\{x_k\} \subset \Omega$ is bounded, without loss of generality, we assume $x_k \longrightarrow x^*$. By (2.1) and (2.8), we have

$$\lim_{k\to\infty} d_k = -\lim_{k\to\infty} g_k + \lim_{k\to\infty} \beta_k^{H*} d_{k-1} - \lim_{k\to\infty} \theta_k^{H*} y_{k-1} = -\nabla f(x^*), \tag{3.23}$$

where we use (3.18), (2.10) and the fact that the sequence $\{d_k\}$ is bounded.
On the other hand, we have

$$\lim_{k\to\infty} \nabla f(x_k + \delta_k \alpha_k^{'} d_k) = \nabla f(x^*). \tag{3.24}$$

Hence, from (3.21)-(3.24), we obtain

$$-\theta_k \nabla f(x^*)^T \nabla f(x^*) \geq 0, \tag{3.25}$$

which means $||\nabla f(x^*)|| = 0$. This contradicts with (3.11). The proof is completed.

4 Numerical results

In this section, we compared the performance of our method with the Convergence properties of an iterative method for solving symmetric nonlinear equations [7]. For the both th algorithms the following parameters are set to $\omega_1 = \omega_2 = 10^{-4}$, $\alpha_0 = 0.01$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^2}$.

The codes for both methods were written in Matlab 7.4 R2010a and run on a personal computer 1.8 GHz CPU processor and 4 GB RAM memory. We stopped the iteration if the toatal number of iterations exceeds 2000 or $||F_k|| \leq 10^{-4}$. "-" to represents failure due to; (i) Memory requirement (ii) Number of iteration exceed 2000 (iii) If $||F_k||$ is not a number. The methods

Table 1. Problem 1

		Algorithm		CPIM	
Dimension	*Guess*	*iter*	*Time*	*iter*	*Time*
500	x_1	47	1.904618	59	2.720469
	1_2	44	1.906971	58	2.611969
	x_3	29	0.357841	55	0.821003
	x_4	26	0.325717	58	0.852849
1000	x_1	30	3.892519	59	8.44528
	x_2	46	5.123549	57	7.371416
	x_3	45	5.079378	57	6.675981
	x_4	23	2.681203	59	6.413456
10000	x_1	47	423.2075	58	531.2987
	x_2	34	296.6762	57	565.5779
	x_3	27	195.2569	57	516.8368
	x_4	62	624.3007	58	548.0929

Table 2. Problem 2

		Algorithm		CPIM	
Dimension	*Guess*	*iter*	*Time*	*iter*	*Time*
500	x_1	11	0.114407	44	0.162467
	x_2	-	-	-	-
	x_3	13	0.04339	20	0.078566
	x_4	13	0.043487	44	0.1407
1000	x_1	14	0.073205	48	0.229995
	x_2	-	-	-	-
	x_3	16	0.0836	27	0.123703
	x_4	14	0.069656	48	0.225926
10000	x_1	16	0.545201	62	2.045299
	x_2	-	-	-	-
	x_3	14	0.502932	11	0.607766
	x_4	16	0.499957	61	1.984683
100000	x_1	11	3.803612	4	2.188995
	x_2	-	-	-	-
	x_3	8	2.838421	-	-
	x_4	11	3.159931	-	-

Table 3. Problem 3

		Algorithm		CPIM	
Dimension	*Guess*	*iter*	*Time*	*iter*	*Time*
1000	x_1	20	0.197767	24	0.244292
	x_2	16	0.146626	23	0.208834
	x_3	-	-	-	-
	x_4	16	0.171042	-	-
10000	x_1	12	2.585215	33	4.466799
	x_2	20	2.409224	9	1.592982
	x_3	-	-	-	-
	x_4	30	6.443075	-	-

were tested on some Benchmark test problems with different initial points. Problem 1 and 2 are from [13] while the remaining one is an artifitial problem.

Problem 1

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & 0 & 2 \end{pmatrix} x + (sinx_1 - 1, \ldots, sinx_n - 1)^T$$

Problem 2. *The discretized Chandrasehar's H-equation:*

$$F_i(x) = x_i - (1 - \frac{c}{2n} \sum_{j=1}^{n} \frac{\mu_i x_j}{\mu_i + \mu_j})^{-1}, \quad for i = 1, 2, \ldots, n,$$

wth $c \in [0, 1)$ and $\mu = \frac{i-0.5}{n}$, for $1 \leq i \leq n$. (In our experiment we take $c = 0.9$).

Problem 3.*The Singular function:*

$$F_1(x) = \frac{1}{3}x_1^3 + \frac{1}{2}x_2^2$$
$$F_i(x) = -\frac{1}{2}x_i^2 + \frac{i}{3}x_i^3 + \frac{1}{2}x_{i+1}^2, \quad i = 2, 3, \ldots, n-1$$
$$F_n(x) = -\frac{1}{2}x_n^2 + \frac{n}{3}x_n^3$$

The tables listed numerical results, where "Iter" and "Time" stand for the total number of all iterations and the CPU time in seconds, respectively;$||F_k||$ is the norm of the residual at the stopping point.The numerical results indicate that the proposed Algorithm compared to IPRP has minimum number of iteration and CPU time respectively. Also $x_1 = (1, 1, \ldots, n)$, $x_2 = (0, 0, \ldots, 0)$, $x_3 = (1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n})$ and $x_4 = (1 - 1, 1 - \frac{1}{2}, 1 - \frac{1}{3}, \ldots, 1 - \frac{1}{n})$.

5 Conclusion

In this paper, an efficient hybrid conjugate gradient method for solving large-scale symmetric nonlinear equations is derived. It is a fully derivative-free iterative method which possesses global convergence under some reasonable conditions. Numerical comparisons using a set of large-scale test problems show that the proposed method is promising.

References

[1] W. Cheng and Z. Chen, Nonmonotone Spectral method for large-Scale symmetric nonlinear equations,*Numer. Algorithms*, **62**, 62149-162 (2013).

[2] G. Gu, D. Li, L. Qi and S.-Z. Zhou, Descent direction of quasi-Newton methods for symmetric nonlinear equations, *SIAM J. Numer. Anal.*, **40**, 1763-1774 (2002).

[3] L. Zhang, W. Zhou, and D.-H. Li, Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, *Numer. Math.*, **104**, 561-572 (2006).

[4] D.H. Li and M. Fukushima, A globally and superlinearly convergent Gauss-Newton-based BFGS methods for symmetric nonlinear equations, *SIAM J. Numer. Anal.*, **37**, 152-172 (1999).

[5] D.-H. Li and X Wang, A modified Fletcher Reeves-type derivative-free method for symmetric nonlinear equations, *Numer. Algebra Control Optim.*, **1**, 71-82 (2011).

[6] W. Zhou, A globally and R-linearly hybrid HS and PRP method and its inexact version with applications, *Numer. Math.*, **104**, 561-572 (2006).

[7] W. Zhou and D. Shen, *Convergence properties of an iterative method for solving symmetric nonlinear equations* , *J. Optim. Theory Appl.*, doi: 10. 1007/s10957-014-0547-1 (2014) .

[8] M.Y. Waziri and J. Sabiu, An alternative conjugate gradient approach for large-scale symmetric nonlinear equations, *Journal of mathematical and computational science*, **6**, 855-874 (2016).

[9] X Yunhai, W. Chunjie and Y.W. Soon, Norm descent conjugate gradient method for solving symmetric nonlinear equations, *J. Glo. Optim.*, DOI 10.1007/s10898-014-0218-7 (2014).

[10] G. Yuan, X. Lu and Z. Wei, BFGS trust-region method for symmetric nonlinear equations, *J. Comput. Appl. Math.*, **230**, 44-58 (2009).

[11] J. Sabi'u, Enhanced derivative-free conjugate gradient method for solving symmetric nonlinear equations *International Journal of Advances in Applied Sciences*, **5**, 1 (2016).

[12] J. sabi'u and U. Sanusi, An efficient new conjugate gradient approach for solving symmetric nonlinear equations , *Asian Journal of Mathematics and Computer Research*, **12**, 34-43 (2016).

[13] M.Y. Waziri and J. Sabi'u, A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations, *International J. of mathematics and mathematical science*, doi:10.1155/2015/961487 (2015).

[14] W.W. Hager and H. Zhang, A New conjugate gradient Method with Guaranteed Descent and an efficient line search, *SIAM J. Optim.* **16**, 170-192 (2005).

Author information

Jamilu Sabi'u, Department of Mathematics, Northwest University, Kano, Nigeria.
E-mail: sabiujamilu@gmail.com

Received: April 4, 2016.

Accepted: September 20, 2016.