A new class of three-term double projection approach for solving nonlinear monotone equations

M M Mahdi$^1$ and Mushtak A.K. Shiker$^{1,2}$

$^1$Department of Mathematics, College of Education for Pure Sciences, University of Babylon, Babil-Iraq
E-mail: $^1$mohmath44@gmail.com, $^2$mmtmmhh@yahoo.com

$^2$Corresponding Author

Abstract. The derivative-free projection methodology is important and highly efficient method to solve large scale monotone equations of nonlinear systems. In this work, we suggested a new class of extensions projection approach employs along with a new line search to show a class of new double projection technique for solving monotone systems of nonlinear equations. Our algorithm can be applied to solve nonsmooth equations, furthermore it's suitable for large scale equations due to simplicity and limited memory. This method constrains new two appropriate hyperplanes in each point strictly separates $x_k$ from the solution set, it can obtain the next iteration $x_{k+1}$ by projecting $x_k$ onto the intersection of two halfspaces and include the solution set of the problem. The global convergence of the given method is investigated with mild assumptions. The numerical experiments prove that the new approach is working well and so promising.

Keywords: Double Projection Algorithm, Monotone Equations, Line Search Method and Conjugate Gradient Descent.

1. Introduction
The derivative-free projection techniques are the most effective line search methods to solve nonlinear systems of equations. Consider the following systems of equations

$$F(x) = 0, \quad x \in \mathbb{R}^n, \quad (1.1)$$

s.t. $F: \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$ be continuous and nonlinear function, $\Omega \neq \emptyset$ closed convex, by monotonicity we mean

$$\langle F(x) - F(y), x - y \rangle \geq 0, \quad \text{for all } x, y \in \mathbb{R}^n.$$  

The gradient projection techniques are efficient to find the solution of large scale unconstraint optimization equations because it inherit nice properties of conjugate gradient descent method such as a low storage require and high efficient. A lot of computation methods have been proposed to solve unconstraint nonlinear problems. For example, Newton method, quasi newton method and Levenberg-Marquardt type method and other methods, see [1, 2, 3, 4]. These methods have a particular important because they can hold a properties of convergence easily for any a suitable firstly guesses but they have some disadvantages because these methods fail when deal with a large scale cases, since they need, when works, to compute the Jacobean matrix or an approximation of it to solve a linear system of equations. Projection method is good approach to solve large scale equations [5, 6, 7, 8]. A good property of the derivative-free projection technique for solving the monotone equation is that competitive with conjugate gradient descent [9]. In this work, we developed a new class of three terms of a derivative-free with a monotone line search technique. Motivated by the idea of Yuan [6],...
we constructed a new projection method of three terms derivative-free technique for solving a nonlinear systems of equations. Meanwhile, we extension this projection technique to generate a double projection algorithm which has a good convergence property and it is more efficient than classical projection method. The rest of this paper is as follows, in section two, we consider a projection approach and some preliminaries and we prove its sufficient descent property. In section three: we introduce the framework of our algorithm. In section four, the global convergence is established. Finally, in section five, the numerical results and performance profile of this method and finally, some conclusion of our work.

2. Preliminaries of Projection

Given this section, we consider a projection approach that applied to solve unconstraint optimization problem

\[
\min_{x \in \mathbb{R}^n} f(x),
\]

were \( F: \mathbb{R}^n \to \mathbb{R} \) is continuously and differentiable. For the projection method proposed by Solodov and Svaiters [5], firstly determine an initial point \( x_0 \), an iterative scheme for (1.1) generates a sequence \( \{x_k\} \) by \( x_{k+1} = x_k + \alpha_k d_k, k \in \mathbb{N}, \) where a line search procedure employs along the direction \( d_k \) to calculate step size \( \alpha_k \) such that

\[
\langle F(z_k), (x - z_k) \rangle > 0.
\]  

(2.1)

where \( z_k = x_k + \alpha_k d_k \). On the other hand, using the monotonicity of \( F \), for any \( \hat{x} \) s.t. \( F(\hat{x}) = 0 \), then

\[
\langle F(z_k), (\hat{x} - z_k) \rangle \leq 0.
\]  

(2.2)

The hyperplane

\[
H_k = \{ x \in \mathbb{R}^n | F(z_k)^T (x - z_k) = 0 \},
\]

strictly separates the current iterate from zeros of the problem (1.1). The other iteration \( x_{k+1} \) is constructed by projecting \( x_k \) onto \( H_k \) that is \( x_{k+1} \) is determined by:

\[
x_{k+1} = x_k - \frac{\langle F(z_k), (x - z_k) \rangle}{\|F(z_k)\|^2} F(z_k).
\]  

(2.3)

Many authors proposed the projection method for monotone equations with the other line search and direction, see [2, 6, 10, 11, 12, 13]. The authors introduced many techniques to solve various optimization and reliability problems (see 14- 21), but in this work, we introduce a new idea of a double projection technique based on a classical projection technique to solve nonlinear systems of large scale equations. The terminology of suggested algorithm, at each approximate \( x_k \) the suggested algorithm generates two appropriate hyperplanes which strictly separate \( x_k \) from the solution set of (1.1) and then obtains the new iteration by projecting \( x_k \) onto the intersection of two halfspaces that are generated from above hyperplanes which contains all solutions of (1.1). We note that the sequence of the distances be decreasing always from the approximation of the solution set of problems, iterates determined by this technique are converge to the solution. This approach does not contain any derivative and it is good for solving nonsmooth equations. Also, the new algorithm establishes the properties of convergence even when the zeros of problem is not singleton and it gives a good results and so promising.

3. A New Double-Projection

In this part, based on the originally projection approach with a monotone equations introduced by zhang and zhou [11], we suggest a class of double projection approach with an a suitable backtracking line search strategy along the search direction \( d_k \). We will propose the following a new direction formula for nonlinear monotone equations (1.1)

\[
d_k = \begin{cases} 
-F_k + \beta_k \psi_{k-1} + \delta_k y_{k-1}, & k \geq 1 \\
-F_k & k = 0
\end{cases}
\]

(3.1)

where
\[ s_k = x_k - x_{k-1}, \quad y_k = F_k - F_{k-1} + \xi s_k, \quad \beta_k^{\text{DMO}} = \frac{\mu F_k^T s_k}{\max \{ y \| s_k \| y_k, -d_k^T F_k \}}, \]
\[ \delta_k = \frac{-\mu^T y_k}{\max \{ y \| s_k \| y_k, -d_k^T F_k \}}, \quad \gamma, \xi \in (0,1), \quad \mu > 0. \]

Let \( x_k \) be a present approximate, the framework of the suggested method is that: using the search direction \( d_k \) in (3.1) with the line search method to generate a new nonnegative step length \( \alpha_k \) that satisfy (2.1). Then the new algorithm generates appropriate hyperplane \( H_k \) that strictly separates \( x_k \) from zeroes of problem (1.1). We determined the point \( x_k^* \) by projecting \( x_k \) onto \( H_k \) such that
\[ x_k^* = x_k - \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k)\|^2}, \quad F(z_k^*) > 0, \quad (3.2) \]
where \( z_k^* = x_k^* + \alpha_k^* d_k^* \).

And the hyperplane
\[ H_k^* = \{ x \in \mathbb{R}^n \mid F(z_k^*)^T (x - z_k^*) = 0 \}, \quad (3.3) \]
strictly separates \( x_k^* \) from the solution set of problem (1.1). By letting
\[ G_k = \{ x \in \mathbb{R}^n \mid F(z_k)^T (x - z_k) \leq 0 \text{ and } F(z_k^*)^T (x - z_k^*) \leq 0 \}, \]
it is obvious that \( G_k \) contains all solutions of problem (1.1), but \( x_k, x_k^* \not\in G_k \). By projecting \( x_k \) onto \( G_k \), we can obtain the best iteration for a solution of system (1.1).

So, the next iteration \( x_{k+1} \), is computed by:
\[ x_{k+1} = P_k(x_k) = \text{argmin}\{ \|x - x_k\| \mid x \in G_k \}. \quad (3.4) \]

We can calculate \( x_{k+1} \) easily by solving the following subproblem
\[ \min_{x} \frac{1}{2} \|x - x_k\|^2 \quad \text{s.t.} \quad F(z_k)^T x \leq F(z_k)^T z_k \]
\[ F(z_k^*)^T x \leq F(z_k^*)^T z_k^* \]
\[ (3.5) \]
It is easily to see that if the vectors \( F(z_k) \) and \( F(z_k^*) \) are parallel then the initial constraint is not necessary and \( x_{k+1} \) can be calculated by:
\[ x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k^*)\|^2} F(z_k^*), \]
when the vectors \( F(z_k) \) and \( F(z_k^*) \) are not parallel because the objective function of (3.5) is a convex quadratic function and its constrains are linear, the next Karush-Kuhn-Tucker conditions [10] are sufficient for \( x_{k+1} \) to be solution of (3.5):
\[
\begin{cases}
(1) \ x_{k+1} - x_k + \tau_1 F(z_k) + \tau_2 F(z_k^*) = 0, \\
(2) \ \tau_1 (F(z_k)^T x_{k+1} - F(z_k)^T z_k) = 0, \\
(3) \ \tau_2 (F(z_k^*)^T x_{k+1} - F(z_k^*)^T z_k^*) = 0, \\
(4) \ F(z_k)^T x_{k+1} - F(z_k)^T z_k \leq 0, \\
(5) \ F(z_k^*)^T x_{k+1} - F(z_k^*)^T z_k^* \leq 0, \\
(6) \ \tau_1 \geq 0, \\
(7) \ \tau_2 \geq 0.
\end{cases}
\]
\[ (3.6) \]

where \( \tau_1 \) and \( \tau_2 \) are Lagrangian multipliers.

Hence, if \( \tau_1 = \tau_2 = 0 \) then (1) results \( x_{k+1} = x_k \) and this contradicts (4).

If \( \tau_1 > 0, \tau_2 = 0 \), then the conditions (1) and (2) result \( x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k)^T\|^2} F(z_k) = x_k^* \), and this contradicts (5). Now, the next two cases are possible:

- If \( \tau_1 = 0 \) and \( \tau_2 > 0 \), the conditions (1) and (2) conclude
4

\[ x_{k+1} = x_k - \frac{F(z_k^*)^T (x_k - z_k^*)}{\|F(z_k^*)\|^2} F(z_k^*), \]

If \( \tau_1 > 0 \) and \( \tau_2 > 0 \), then the conditions (1), (2) and (3) conclude

\[ \tau_1 \|F(z_k^*)\|^2 + \tau_2 F(z_k^*)^T F(z_k^*) = F(z_k^*)^T (x_k - z_k) \]

\[ \tau_1 F(z_k^*)^T F(z_k^*) + \tau_2 \|F(z_k^*)\|^2 = F(z_k^*)^T (x_k - z_k^*) \]

(3.7)

It is easy to show that the coefficient matrix of system (3.7) is positive by Cauchy-Schwarz inequality, and it is obvious to see that \( \tau_1, \tau_2 \) can be generated by solving (3.7), Thus, solving the subproblem (3.5) is relatively inexpensive to implement.

In our new algorithm, we use the new line search:

\[ -F(x_k + \alpha_k d_k)^T d_k \geq \eta \Gamma \alpha_k \|F(z_k^*)\| \|d_k\|^2, \]

where \( \alpha_k = \max\{\psi^i \psi_k, i = 1,2,3,\ldots\}, \eta > 0, \Gamma, \psi, \nu \in (0,1) \) and \( s_k \) is firstly guess of \( \alpha_k \). The same line search conditions have been found for other techniques in [3, 10, 11]. Now, we can state the proposed algorithm to solve (1.1).

3.1 A New Double Projection Algorithm (DMO):

0. The initial point \( x_0 \in R^n \) is given, parameters \( \eta, \xi, \mu > 0 \) and \( \psi, \gamma, \epsilon, \nu \in (0,1) \).

Let \( k = 0, \quad F_0 = -F(x_0) ; \quad d_0 = -F_0; \quad \|F_k\| > \epsilon; \)

1. (The First Step Length)

Calculate the direction \( d_k \) by (3.1). Take an initial step length \( \alpha_k \) s.t. \( \alpha_k = \psi_k; \)

The line search of this method is determined by (3.8) and \( \alpha_k = \max\{\psi^i \psi_k, i = 1,2,3,\ldots\}; \)

Compute \( \gamma = x_0 + \alpha_k d_k \).

2. (The Trial Point)

If \( \|F(x_k^*)\| \leq \epsilon, \quad x_{k+1} = x_k; \)

else calculate \( x_k^* \) by (3.1);

Compute \( \gamma = F(x_k^*); \)

If \( \|F(x_k^*)\| \leq \epsilon, \quad x_{k+1} = x_k^*; \)

Stop;

3. (The Second Steplength)

Calculate the direction \( d_k \) by

\[ d_k = \begin{cases} -F_k^* + \beta_k^{DMO} y_k - s_k^* s_k^* - d_k^*, & k \geq 1 \\ -F_k^* & k = 0 \end{cases} \]

where

\[ s_k^* = x_k^* - x_k, \quad y_k^* = F_k^* - F_k^* - \xi s_k^* , \]

\[ \beta_k^{DMO} = \frac{\mu F_k^* s_k^*}{\max\{y_k^* s_k^* \| \| y_k^* s_k^* \| \| -d_k^* F_k^* \}} \]

\[ \delta_k = \frac{\max\{y_k^* s_k^* \| \| y_k^* s_k^* \| \| -d_k^* F_k^* \}}{\max\{y_k^* s_k^* \| \| y_k^* s_k^* \| \| -d_k^* F_k^* \}} \]

Set \( \alpha_k = \psi_k \). Compute the line search by

\[ -F(x_k^* + \alpha_k^* d_k)^T d_k^* \geq \eta \Gamma \alpha_k^* \|F(x_k^* + \alpha_k^* d_k)\| \|d_k^*\|^2. \]

Thus \( z_{k+1}^* = x_k^* + \alpha_k^* d_k^*; \)

4. (The New Point Calculation)

If \( \|F(z_{k+1}^*)\| \leq \epsilon, \quad x_{k+1} = z_{k+1}^*; \)

stop;

Calculate \( x_{k+1} \) by solving (3.5);

\( F_{k+1} = F(x_{k+1}); \)

Let \( k = k + 1, \) and go to step 1;
3.2 Remark: It is easy to see from algorithm (3.2) that
\[-F_k^T d_k \geq \|F_k\|^2 \]  
(3.9)

4. Global Convergence
In this part, we investigate the global convergence of the offer approach and we used some necessary assumptions (B):

B₁. The solution set of nonlinear equations (1.1) is nonempty.
B₂. The mapping \( F(x) \) is monotone and Lipschitz continuous of \( R^n \), i.e., \( \exists L > 0 \), such that
\[ \|F(x) - F(y)\| \leq L \|x - y\| \] for all \( x, y \in R^n \).  
(4.1)
The projection operator [10] is a mapping \( P_\Omega: R^n \to \Omega \) for all \( x \in R^n \) where \( \Omega \) is a nonempty closed convex subset.
\[ P_\Omega(x) = \text{argmin}\{\|x - z\| | z \in \Omega\} \]

We used the properties of projection operator to analyze the convergence properties of DMO algorithm in the next lemma.

4.1 Lemma [7]: Let \( \Omega \subset R^n \) be a nonempty closed convex set and \( P_\Omega(x) \) be the projection of \( x \) onto \( \Omega \) for all \( x, y \in R^n \) the following steps hold:
I. For any \( z \in \Omega, \langle P_\Omega(x) - x, z - P_\Omega(x) \rangle \geq 0 \).
II. \( \langle P_\Omega(x) - P_\Omega(y), x - y \rangle \geq 0 \). The inequity is strict when \( P_\Omega(x) \neq P_\Omega(y) \).
III. \( \|P_\Omega(x) - P_\Omega(y)\| \leq \|x - y\| \).  
□
Based on the next lemma, we note that from the approximation to the solution set of problem decreases with \( k \), this interesting property to DMO algorithm.

4.2 Lemma (4.2): Let assumptions (B₁, B₂) satisfied. For the sequence \( \{x_k, z_k\} \) which is determined by algorithm DMO, for any solution \( \bar{x} \) of (1.1) we get
\[ \|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2. \]  
(4.2)
And the sequence \( \{x_k\} \) is bounded. Moreover, either the sequence \( \{x_k\} \) is finite while the last iterate is a solution or the sequence is infinite and
\[ \lim_{k \to \infty} \|x_{k+1} - x_k\| = 0, \]  
(4.3)

Proof: From the definition of \( G_k \), it is easily to know that \( G_k \) is a closed convex set and \( \bar{x} \in G_k \). This result together with (3.4) and the first part (I) of Lemma (4.1) conclude that
\[ (x_k - x_{k+1})^T (x_{k+1} - \bar{x}) \geq 0, \]
where
\[ \|x_k - \bar{x}\|^2 = \|x_k - x_{k+1} + x_{k+1} - \bar{x}\|^2 \]
\[ = \|x_k - x_{k+1}\|^2 + 2(x_k - x_{k+1}, x_{k+1} - \bar{x}) + \|x_{k+1} - \bar{x}\|^2 \]
\[ \geq \|x_k - x_{k+1}\|^2 + \|x_{k+1} - \bar{x}\|^2. \]
This means
\[ \|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \|x_{k+1} - x_k\|^2. \]  
(4.4)
From (4.4), the sequence of \( \|x_k - \bar{x}\| \) is a descent sequence. Theorem (2.1) in Solodov and Svaiter’s [5] gave the remaining results.  
□
In the following lemma we prove that the sequence \( \{d_k\} \) is bounded. This property interesting to holds properties of convergence to DMO algorithm.

4.3 Lemma: Assume that a condition of lemma (4.1) satisfied, then the sequence \( \{d_k\} \) is bounded i.e. this mean \( \exists \mu, \gamma > 0 \) s.t.
\[ \|d_k\| \leq \left(1 + \frac{2\mu}{\gamma}\right)\|F_k\|, \]  
(4.5)
and \( \forall k \geq 0 \), also we have
\[
\lim_{k \to \infty} \alpha_k \|d_k\| = 0 \quad (4.6)
\]

**Proof:** by lemma (4.1) we know that the sequence \( \{x_k\} \) is bounded. This result and by continuous mapping \( F \) gives that the sequence of \( \{F(x_k)\} \) is also bounded, so

\[
\|d_k\| \leq \|F_k\| + \beta^\text{DMO} \|y_{k-1}\| + \|\delta_k\|s_{k-1}\|s_k\| + \mu \|y_k\|\|F_k\|.
\]

This result \( \|d_k\| \) is bounded.

By (3.1) and (3.4), it can be conclude that

\[
\|x_{k+1} - x_k\| \geq \|x_k - x_k^*\| \quad (4.7)
\]

On the other side, from the definition of \( x_k^* \) proposed in (3.1), we have

\[
\|x_k - x_k^*\| = \frac{\|F(z_k)^T(x_k - z_k)\|}{\|F(z_k)\|} = -\alpha_k \|F(z_k)\|d_k.
\]

So, by (3.8) with (4.7) together (4.8) conclude that

\[
\|x_{k+1} - x_k\| \geq \frac{\alpha_k \|F(z_k)\| \|d_k\|^2}{\|F(z_k)\|}.
\]

From the boundedness of sequence \( \{x_k\} \) and \( \{d_k\} \), we get

\[
\|x_{k+1} - x_k\| = \alpha_k^2 \eta \Gamma \|d_k\|^2 \geq 0.
\]

The above inequality with (4.3) gives us (4.6). And the proof is complete. \( \square \)

In the next Lemma we show that the line search of proposed algorithms is well-defined.

**4.4 Lemma:** Let assumptions \( (B_1, B_2) \) satisfied, and the sequence \( \{x_k\} \) is determined by DMO algorithm implies that the line search of steps (1 and 3) in DMO algorithm is well-defined.

**Proof:** Firstly, we will show that step 1 in our algorithm is well defined, and by a contradiction, assume that \( \hat{k} \) is iteration index, the condition (3.8) does not true. So, that if we take \( \alpha_k^m = v^m \psi_k \)

\[
-F(x_k + \alpha_k^m d_k)^T d_k \geq \eta \Gamma \alpha_k^m \|F(x_k + \alpha_k^m d_k)\|d_k\|^2 \quad (4.9)
\]

By the definition of search direction \( d_k \), (4.5) and assumptions B2 conclude that

\[
\|F(x_k)\|^2 = -F(x_k)^T d_k = (F(x_k + \alpha_k^m d_k) - F_k)^T d_k - F(x_k) + \alpha_k^m d_k^T d_k \\
< \eta \Gamma \alpha_k^m \|F(z_k)\|\|d_k\|^2 = \eta \Gamma \alpha_k^m \|F(z_k)\|\|d_k\|^2 = (L + \eta \Gamma \|F(z_k)\|) \alpha_k^m \|d_k\|^2.
\]

Thus, it follows from lemma (4.2) and (4.6) that the sequence \( \{x_k\} \) is bounded. Let \( M > 0 \), satisfies \( \|F(x_k)\| \leq M \), then from (3.1) we get

\[
\|F(x_k + \alpha_k^m d_k)\| \leq \|F(x_k + \alpha_k^m d_k) - F_k\| + \|F_k\| \\
\leq \Lambda \|d_k\| + M \\
\leq LM \psi \left(1 + \frac{2\mu}{\gamma}\right) + M.
\]

Thus, \( \forall m \geq 0 \), we have

\[
\alpha_k^m > \frac{\|F_k\|^2}{(L + \eta \Gamma \|F(z_k)\|)\|d_k\|^2} = \frac{M^2}{(L + \eta \Gamma LM \psi \left(1 + \frac{2\mu}{\gamma}\right) + M) \left(M + \frac{2\mu M}{\gamma}\right)^2} > 0.
\]

The above inequality generates a contradiction with the definition of \( \alpha_k^m \). Hence, the step 1 of the DMO algorithm is well-defined. In the same way, we resulted that step 3 is well-defined too. And the proof is complete. \( \square \)

**4.5 Theorem:** Assume that any conditions in lemma (4.1) are satisfied, then

\[
\lim_{k \to \infty} \inf \|F_k\| = 0 \quad (4.10)
\]

**Proof:** Suppose that (4.10) is not true. Then \( \exists \epsilon > 0 \) s.t
\[ \| F_k \| \geq \epsilon, \forall k \geq 0. \]  \hfill (4.11)

This conclude
\[ \| d_k \| \geq \epsilon_1, \quad \forall k \geq 0. \text{ where } \epsilon_1 = (1 + \frac{2\mu}{\gamma})\epsilon \]

So, it follows from (4.6) that
\[ \lim_{k \to \infty} \alpha_k = 0. \]

By the first step of DMO algorithm it’s clear that \( \alpha_k = \nu^{-1} \alpha_k \) does not hold (3.8) i.e.
\[ -F(x_k + \alpha_k d_k)^T d_k < \eta \Gamma \alpha_k \| F(x_k) \| \| d_k \|^2. \]  \hfill (4.12)

Since the sequence \( \{ x_k \} \) is a bounded in lemma (4.2), \( \exists \bar{x}, \bar{\alpha} \) is a limit point and the index set \( K_1 \) is an infinite this mean \( \lim_{k \to \infty} x_k = \bar{x} \) for \( k \in K_1 \). At the same time, and by lemma (4.1), \( \exists K_2 \subset K_1 \) and \( K_2 \) an infinite index set and an a limit point \( \bar{d} \) s. t. \( \lim_{k \to \infty} d_k = \bar{d} \) for \( k \in K_2 \). Consider that
\[ F(x_k)^T d_k = -\| F_k \|^2 \leq -\epsilon^2 < 0 \]

Thus we take \( \lim_{k \to \infty} \) in two terms of (4.12) for \( k \in K_2 \), it is gives
\[ F(x^*)^T d^* \leq -\epsilon^2 < 0. \]  \hfill (4.13)

On the other side, taking \( \lim_{k \to \infty} \) in two terms of (4.13) for all \( k \in K_2 \), we obtain
\[ F(\bar{x})^T \bar{d} \geq 0. \]  \hfill (4.14)

Thus implies a contradiction with (4.13) hence (6.22) holds. The proof is complete. \( \square \)

5. Numerical Experiments

In this part, we report the numerical experiments to assess the competitiveness and robustness of DMO method to solve nonlinear systems of monotone equations. We compare our algorithm with a three famous methods to obtain the best numerical results. The DMO algorithm compared with the following algorithms:

- **HS**: A projection Hestenes-Stiefel like method comes from Awwal et al. [1].
- **SDA**: A scaled derivative-free projection method comes from Koorapetse et al. [3].
- **NNT**: Some three-terms conjugate gradient methods comes from Liu et al. [10].

We implement all codes in each approach using MATLAB R2014a and run PC with 4GH, CPU2.30-Core i5 Windows 8 operation system. The breaking criteria of all method when the total number of iteration exceeds 500000 or \( \| F_k \| \leq 10^{-8} \) or \( \| F(z_k) \| \leq 10^{-8} \). The parameters of new method were set as follows: \( \gamma = 0.7, \Gamma = 0.001, \beta = 0.03, \eta = 0.001, \nu = 0.7, \psi = 0.001, \mu = 2 \) and \( \xi = 0.001 \). The parameter of other methods comes from [1], [3] and [10] respectively. The starting points of all problems are initialized similar in [7] as following:

\[
x_0 = (10,10,...,10)^T, \quad x_1 = (-10,-10,...,-10)^T, \quad x_2 = (1,1,...,1)^T, \quad x_3 = (-1,-1,...,-1)^T, \quad x_4 = (1,\frac{1}{2},\frac{1}{3},...\frac{1}{n})^T, \quad x_5 = (0.1,0.1,...,0.1)^T, \quad x_6 = (\frac{1}{n},\frac{1}{n},...\frac{1}{n})^T, \quad x_7 = (1-\frac{1}{n},1-\frac{2}{n},...0)^T.
\]

The comparison has been based on the number of iteration \( (N_i) \), number of functions evaluations \( (N_f) \) and CPU time with the dimensions limited to 5000 - 50000. In table 1, we reported the numerical results of the iterations and functions evaluations taken for each approach to satisfy the optimal value. While, in table 2 the numerical experiment represent the CPU time for each method. We note that our method DMO works better than other methods in solving these problems. We see that DMO algorithm in some problems performs a little lower than NNT algorithm but still very-well as compared in general with the three methods.
Table 1: Numerical results

| P. | Dim. | S.P | DMO | HS | NNT | SDA |
|----|------|-----|-----|----|-----|-----|
|    |      |     |     |    |     |     |
|    |      |     |     |    |     |     |
| P1 | 20000 | x₀ | 16  | 122| 121| 244| 60  | 183| 261| 523|
|    | 20000 | x₁ | 16  | 122| 121| 244| 60  | 183| 261| 523|
|    | 20000 | x₂ | 19  | 120| 112| 226| 52  | 158| 269| 539|
|    | 20000 | x₃ | 19  | 120| 112| 226| 52  | 158| 269| 539|
|    | 20000 | x₄ | 12  | 78 | 118| 303| 33  | 101| 170| 341|
|    | 20000 | x₅ | 16  | 102| 95 | 192| 45  | 137| 228| 457|
|    | 20000 | x₆ | 18  | 114| 86 | 200| 51  | 155| 260| 521|
|    | 20000 | x₇ | 18  | 114| 78 | 189| 51  | 155| 260| 521|
| P2 | 50000 | x₀ | 106 | 1086| 121| 244| 60  | 183| 304| 611|
|    | 50000 | x₁ | 2   | 60 | 41 | 84 | 17  | 54 | 89 | 180|
|    | 50000 | x₂ | 19  | 120| 112| 226| 52  | 158| 268| 538|
|    | 50000 | x₃ | 130 | 1310| 33 | 68 | 14  | 45 | 92 | 187|
|    | 50000 | x₄ | 12  | 78 | 120| 321| 33  | 101| 169| 340|
|    | 50000 | x₅ | 16  | 102| 95 | 192| 45  | 137| 227| 456|
|    | 50000 | x₆ | 18  | 114| 91 | 209| 51  | 155| 259| 520|
|    | 50000 | x₇ | 18  | 114| 91 | 209| 51  | 155| 259| 520|
| P3 | 50000 | x₀ | 128427| 935570| 32933| 66433| 87692| 350775| 80563| 161129|
|    | 50000 | x₁ | 142607| 1038716| 36768| 74001| 97070| 388287| 89469| 178941|
|    | 50000 | x₂ | 102734| 748492| 26378| 53108| 71994| 287983| 65675| 131353|
|    | 50000 | x₃ | 123804| 901288| 32357| 64741| 84639| 338563| 77670| 155343|
|    | 50000 | x₄ | 107517| 783028| 27718| 55520| 72874| 291512| 66492| 132987|
|    | 50000 | x₅ | 103700| 755004| 26736| 53501| 70452| 281815| 64229| 128461|
|    | 50000 | x₆ | 39252| 287859| 11456| 23043| 30890| 123567| 28210| 56433|
|    | 50000 | x₇ | 39246| 287811| 11567| 23243| 30890| 123567| 28210| 56423|
| P4 | 10000 | x₀ | 139 | 979 | 62 | 127| 77  | 266| 140| 283|
|    | 10000 | x₁ | 258 | 1744| 90 | 183| 82  | 278| 188| 379|
|    | 10000 | x₂ | 138 | 971 | 55 | 113| 75  | 263| 95 | 193|
|    | 10000 | x₃ | 206 | 1394| 76 | 155| 72  | 246| 148| 299|
|    | 10000 | x₄ | 176 | 1190| 68 | 139| 69  | 232| 129| 261|
|    | 10000 | x₅ | 163 | 1096| 72 | 147| 66  | 223| 133| 269|
|    | 10000 | x₆ | 160 | 1103| 66 | 157| 71  | 244| 126| 255|
|    | 10000 | x₇ | 160 | 1103| 65 | 156| 71  | 244| 126| 255|
### Table 1: Numerical results - continued

| P_5 | Dim. | S.P. | DMO | HS | NNT | SDA |
|-----|------|------|-----|----|-----|-----|
|     |      |      | Ni  | Nf | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  |
| 10000 | x_0 | 177 | 1593 | 2344 | 11705 | 175 | 528 | 753 | 1509 |
| 10000 | x_1 | 176 | 1584 | 2386 | 11804 | 175 | 528 | 754 | 1511 |
| 10000 | x_2 | 177 | 1593 | 2498 | 12442 | 175 | 528 | 753 | 1509 |
| 10000 | x_3 | 177 | 1593 | 2377 | 11912 | 175 | 528 | 753 | 1509 |
| 10000 | x_4 | 177 | 1593 | 2534 | 12640 | 175 | 528 | 753 | 1509 |
| 10000 | x_5 | 177 | 1593 | 2529 | 12634 | 175 | 528 | 753 | 1509 |
| 10000 | x_6 | 177 | 1593 | 2509 | 12550 | 175 | 528 | 753 | 1509 |
| 10000 | x_7 | 177 | 1593 | 2444 | 12208 | 175 | 528 | 753 | 1509 |

| P_6 | Dim. | S.P. | DMO | HS | NNT | SDA |
|-----|------|------|-----|----|-----|-----|
|     |      |      | Ni  | Nf | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  |
| 5000 | x_0 | 22  | 90  | 128 | 258  | 61  | 185 | 311 | 624 |
| 5000 | x_1 | 22  | 92  | 133 | 268  | 63  | 191 | 322 | 646 |
| 5000 | x_2 | 20  | 84  | 117 | 236  | 56  | 170 | 283 | 568 |
| 5000 | x_3 | 21  | 88  | 123 | 248  | 59  | 179 | 298 | 598 |
| 5000 | x_4 | 20  | 84  | 121 | 244  | 57  | 173 | 292 | 586 |
| 5000 | x_5 | 20  | 84  | 120 | 242  | 57  | 173 | 291 | 584 |
| 5000 | x_6 | 20  | 84  | 119 | 240  | 57  | 173 | 288 | 578 |
| 5000 | x_7 | 20  | 84  | 119 | 240  | 57  | 173 | 288 | 578 |

| P_7 | Dim. | S.P. | DMO | HS | NNT | SDA |
|-----|------|------|-----|----|-----|-----|
|     |      |      | Ni  | Nf | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  | Ni  | Nf  |
| 50000 | x_0 | 29308 | 175854 | 129477 | 258956 | 12937 | 25878 | 310748 | 621498 |
| 50000 | x_1 | 5   | 53  | 71  | 144  | 3811 | 7629 | 174  | 350  |
| 50000 | x_2 | 29303 | 175824 | 129445 | 258892 | 12939 | 25880 | 310670 | 621342 |
| 50000 | x_3 | 2   | 35  | 54  | 110  | 3860 | 7730 | 122  | 246  |
| 50000 | x_4 | 1692 | 10158 | 7490  | 14982 | 785  | 1572 | 17977 | 35956 |
| 50000 | x_5 | 28745 | 172476 | 126968 | 253938 | 12694 | 25390 | 304721 | 609444 |
| 50000 | x_6 | 28906 | 173442 | 128794 | 257609 | 12805 | 25612 | 306459 | 612920 |
| 50000 | x_7 | 28906 | 173439 | 128634 | 257289 | 12805 | 25612 | 306457 | 612916 |
## Table 2: Numerical results (CPU time)

| P   | P1     | P2     | P3     | P4     |
|-----|--------|--------|--------|--------|
|     | Dim.   | S. P   | DMO    | HS     | NNT    | SDA    |
|     |        |        | CPU time |        |        |        |
| 20000 | $x_0$  | 0.46870 | 2.89062 | 2.25000 | 3.81250 |
| 20000 | $x_1$  | 0.37500 | 2.35937 | 1.96875 | 2.75000 |
| 20000 | $x_2$  | 0.34375 | 1.60937 | 1.65625 | 2.40620 |
| 20000 | $x_3$  | 0.34375 | 1.43750 | 1.57812 | 1.87500 |
| 20000 | $x_4$  | 0.20312 | 1.81250 | 0.96875 | 1.07812 |
| 20000 | $x_5$  | 0.26562 | 1.42187 | 1.39062 | 1.43750 |
| 20000 | $x_6$  | 0.34375 | 1.14062 | 1.4375  | 1.79687 |
| 20000 | $x_7$  | 0.39062 | 0.98437 | 1.26562 | 1.67187 |
| 50000 | $x_0$  | 2.93750 | 2.79687 | 2.26562 | 4.37500 |
| 50000 | $x_1$  | 1.93750 | 0.82812 | 0.70312 | 0.93750 |
| 50000 | $x_2$  | 0.29687 | 2.39062 | 1.1875  | 2.48437 |
| 50000 | $x_3$  | 3.17187 | 0.64062 | 0.2500  | 0.76562 |
| 50000 | $x_4$  | 0.20312 | 2.35937 | 0.7500  | 1.10937 |
| 50000 | $x_5$  | 0.29687 | 2.0625  | 0.93750 | 1.54687 |
| 50000 | $x_6$  | 0.32812 | 2.20312 | 1.10937 | 1.7187  |
| 50000 | $x_7$  | 0.34375 | 2.15625 | 1.07812 | 1.87500 |
| 50000 | $x_0$  | 3631.35937 | 286.39062 | 1053.04687 | 579.59375 |
| 50000 | $x_1$  | 4353.64062 | 345.68750 | 1265.9375 | 649.46875 |
| 50000 | $x_2$  | 3093.00001 | 242.79687 | 945.265625 | 470.57812 |
| 50000 | $x_3$  | 3753.43700 | 297.95312 | 1117.82812 | 556.98437 |
| 50000 | $x_4$  | 3290.82812 | 255.04687 | 958.265625 | 473.28125 |
| 50000 | $x_5$  | 3184.67187 | 246.32812 | 1077.01562 | 461.01562 |
| 50000 | $x_6$  | 1189.71800 | 103.96875 | 523.68750 | 200.20312 |
| 50000 | $x_7$  | 1193.48437 | 109.07812 | 524.43750 | 201.29687 |
| 10000 | $x_0$  | 0.75000 | 0.65620  | 0.57812 | 0.60937 |
| 10000 | $x_1$  | 1.03125 | 0.93750  | 0.5625  | 0.73437 |
| 10000 | $x_2$  | 0.57812 | 0.57812  | 0.75000 | 0.32812 |
| 10000 | $x_3$  | 0.75000 | 0.20312  | 0.68750 | 0.43750 |
| 10000 | $x_4$  | 0.75000 | 0.34375  | 0.46875 | 0.32812 |
| 10000 | $x_5$  | 0.51562 | 0.26562  | 0.46875 | 0.29687 |
| 10000 | $x_6$  | 0.75000 | 0.20312  | 0.51562 | 0.23437 |
| 10000 | $x_7$  | 0.64062 | 0.31250  | 0.37500 | 0.25000 |
Table 2: Numerical results (CPU time) - continued

| P  | Dim. | S.P | CPU time |
|----|------|-----|----------|
|    |      |     | DMO      | HS       | NNT      | SDA       |
| P5 | 10000| $x_0$| 0.59375 | 12.76562 | 1.06250  | 2.45312   |
|    | 10000| $x_1$| 0.56250 | 12.84375 | 0.578125 | 1.50000   |
|    | 10000| $x_2$| 0.50000 | 20.25000 | 0.671875 | 1.37500   |
|    | 10000| $x_3$| 0.59375 | 23.26562 | 0.609375 | 1.20312   |
|    | 10000| $x_4$| 0.57812 | 24.42187 | 0.65625  | 1.00010   |
|    | 10000| $x_5$| 0.59375 | 12.0312  | 0.64062  | 0.85937   |
|    | 10000| $x_6$| 0.59375 | 12.23437 | 0.5000   | 0.78125   |
|    | 10000| $x_7$| 0.60937 | 11.59375 | 0.59375  | 0.78125   |
| P6 | 5000 | $x_0$| 0.62500 | 3.53125  | 3.26562  | 8.87500   |
|    | 5000 | $x_1$| 0.54687 | 3.53125  | 2.515625 | 5.07812   |
|    | 5000 | $x_2$| 0.48437 | 3.04687  | 2.015625 | 4.18750   |
|    | 5000 | $x_3$| 0.51562 | 3.28125  | 1.75000  | 4.18750   |
|    | 5000 | $x_4$| 0.43750 | 3.01562  | 1.65625  | 4.09375   |
|    | 5000 | $x_5$| 0.45312 | 3.43750  | 1.35937  | 4.09375   |
|    | 5000 | $x_6$| 0.51562 | 3.62500  | 1.15625  | 4.09375   |
|    | 5000 | $x_7$| 0.54687 | 3.07812  | 1.28125  | 4.03125   |
| P7 | 50000| $x_0$| 477.8750| 1651.3750| 119.750  | 1802.67180|
|    | 50000| $x_1$| 0.18750 | 1.07812  | 34.29687 | 1.12500   |
|    | 50000| $x_2$| 434.0937| 1682.4375| 118.20312| 1627.96870|
|    | 50000| $x_3$| 0.12500 | 0.70312  | 35.51562 | 0.67187   |
|    | 50000| $x_4$| 24.95312| 97.00001 | 6.87500  | 93.81250  |
|    | 50000| $x_5$| 427.54687| 1624.60937| 116.60937| 1588.15620|
|    | 50000| $x_6$| 429.3750| 1672.00001| 118.95312| 1661.25000|
|    | 50000| $x_7$| 430.15625| 1676.07812| 118.29687| 1887.56250|

To compare the performance of all these algorithms, it based on the performance profile of Dolan and More [4]. The performance plots of each method depending on $N_i$, $N_f$ and CPU time that listed in figures (1), (2) and (3) respectively. It is easy to show from figures (1, 2, 3) that the performance of the DMO method is better than the others and it is robust and promising.
Figure 1: Performance of iteration number

Figure 2: Performance of function evaluations

Figure 3: Performance of the CPU time
6. Conclusions
The new class of derivative-free double projection strategy is very interesting to solve both of the optimization problems and nonlinear systems of equations. The suggested method is suitable for large scale equations due to limited memory. The direction of a new method does not require forcing any surplus computing cost. In our algorithm, we investigated global convergence with some assumptions. The numerical experiment and performance profile proved that the DMO technique is efficient and robust.

7. References

[1] Awwal A M, Kumam P, Abubakar A B and Wakili A 2018, A projection Hestenes-Stiefel like method for monotone nonlinear equations with convex constraints, *Thai Journal of Mathematics*, 16, p. 181-199.

[2] Buhmiler S, Krejić N and Lužanin Z 2010, Practical quasi-Newton algorithms for singular nonlinear systems, *Numer. Algorithm*. 55, p.481–502.

[3] Koorapetse M, Kaelol P and Offen R E 2019, A Scaled Derivative-Free Projection Method for Solving Nonlinear Monotone Equations, *Bulletin of the Iranian Mathematical Society*, Vol. 45, p. 755-770.

[4] Dolan E D and Mor’e J J 2002, Benchmarking optimization software with performance profiles. *Math. Program*. 91, p. 201–213.

[5] Solodov M V and Svaiter B F 1999, A globally convergent inexact Newton method for systems of monotone equations, in: M. Fukushima, L. Qi (Eds.), Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, *Kluwer Academic Publishers*, p. 355-369.

[6] Yuan N A 2017 A derivative-free projection method for solving convex constrained monotone equations, *Science Asia*, 43, p. 195-200.

[7] Zarantonello E H 1971 Projections on convex sets in Hilbert space and spectral theory. In: Zarantonello E H (ed.) Contributions to Nonlinear Functional Analysis. *Academic Press*, New York.

[8] Shiker M A K and Amini K 2018, A new projection-based algorithm for solving a large scale nonlinear system of monotone equations, *Croatian operational research review*, 9(1), p. 63-73. [https://doi.org/10.17535/crorr.2018.0006](https://doi.org/10.17535/crorr.2018.0006).

[9] Wasi H A and Shiker M A K 2020, A new conjugate gradient method for solving large scale systems of monotone equations, “in press”, *International Journal of Advanced Science and Technology*, 29 (4), p. 2303-2314.

[10] Liu J K F, Feng Y M and Zou L M 2018, Some three-term conjugate gradient methods with the inexact line search condition, *calcolo*, 55 (2):16.

[11] Mahdi M M and Shiker M A K 2020, A new projection technique for developing a Liu-Storey method to solve nonlinear systems of monotone equations, “in press”, accepted paper for publication in *IOP Science*, 5th International Scientific Conference for Iraqi Al-Khwarizmi Society, Iraq.

[12] Mahdi M M and Shiker M A K 2020, Three terms of derivative free projection technique for solving nonlinear monotone equations, “in press”, accepted paper for publication in *IOP Science*, 5th International Scientific Conference for Iraqi Al-Khwarizmi Society, Iraq.

[13] Mahdi M M and Shiker M A K 2020, Solving systems of nonlinear monotone equations by using a new projection approach, “in press”, accepted paper for publication in *IOP Science*, 1st International Virtual Conference of the University of Babylon (IVCUB)- Iraq.

[14] Mahdi M M and Shiker M A K 2020, Three-term of new conjugate gradient projection approach under Wolfe condition to solve unconstrained optimization problems, “in press”,
accepted paper for publication in *Journal of Advanced Research in Dynamical and Control Systems*.

[15] Hussein H A and Shiker M A K 2020, A modification to Vogel’s approximation method to solve transportation problems, “in press”, accepted paper for publication in *IOP Science, 5th International Scientific Conference for Iraqi Al-Khwarizmi Society, Iraq*.

[16] Hussein H A Shiker M A K and Zabiba M S M 2020, A new revised efficient of VAM to find the initial solution for the transportation problem, “in press”, accepted paper for publication in *IOP Science, 5th International Scientific Conference for Iraqi Al-Khwarizmi Society, Iraq*.

[17] Hussein H A and Shiker M A K 2020, Two New Effective Methods to Find the Optimal Solution for the Assignment Problems, “in press”, accepted paper for publication in *Journal of Advanced Research in Dynamical and Control Systems*.

[18] Shiker M A K and Sahib Z 2018, A modified trust-region method for solving unconstrained optimization. *Journal of Engineering and Applied Sciences*, 13(22), p. 9667-9671. [https://doi.org/10.3923/jeasci.2018.9667.9671](https://doi.org/10.3923/jeasci.2018.9667.9671).

[19] Dwail H H Mahdi M M Wasi H A Hashim K H Dreeb N K, Hussein A H and Shiker M A K (2020), A new modified TR algorithm with adaptive radius to solve a nonlinear systems of equations, “in press”, accepted paper for publication in *IOP Science, 1st International Virtual Conference of the University of Babylon (IVCUB)- Iraq*.

[20] Dwail H H and Shiker M A K 2020, Using a new trust region algorithm with nonmonotone adaptive radius for solving nonlinear systems of equations, “in press”, *International Journal of Advanced Science and Technology*, V 29(4), p 2351- 2360.

[21] Hassan Z A H H and Shiker M A K 2018, Using of generalized baye’s theorem to evaluate the reliability of aircraft systems, *Journal of Engineering and Applied Sciences*, (Special Issue13), p. 10797-10801. [https://doi.org/10.36478/jeasci.2018.10797.10801](https://doi.org/10.36478/jeasci.2018.10797.10801).