IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

Modified Arithmetic Algorithm Based New Conjugate Gradient Method

Ghufran Myasar*, Ban Ahmed

Department of Mathematics, College of Computer Sciences and Mathematics, University of Mosul, Iraq

https://doi.org/10.54153/sjpas.2025.v7i2.1082

Article Information

Received: 25/11/2024
Revised: 29/12/2024
Accepted: 18/01/2025
Published: 30/06/2025

Keywords:

Optimization, Arithmetic Optimization Algorithm, Conjugate Gradient Methods and Meta-Heuristic Algorithms.

Corresponding Author

E-mail: ghofran.23cspq4@student.uomosul.edu.iq
Mobile: +9647701788487

Abstract

In this paper, a new hybrid algorithm is proposed, combining the Arithmetic Optimization Algorithm (AOA), which is a meta-heuristic algorithm that utilizes the distribution behaviour of basic arithmetic operations in mathematics, such as multiplication (M), division (D), subtraction (S), and addition (A), with the classical Conjugate Gradient Algorithm (CGA). The characteristics of CGA are used to enhance the primary population, which is randomly generated as the initial population for the AOA algorithm. The results of the hybrid algorithm are significantly better than those of the original algorithm. Through this hybrid approach, optimal solutions are achieved for most of the functions, with minimum values obtained for these functions. A comparison between the original and hybrid algorithms demonstrates that the hybrid algorithm outperforms the original. Six functions were used, with comparisons made at 500 and 1000 iterations.

Introduction:

One of the main principles of our time is the search for the optimal method or the ideal way to reach an ideal system. Optimization is considered one of the oldest sciences that was established by the researcher Yuqihe, a professor at Harvard University and a member of the American National Academy of Engineering, saying (Optimization is the cornerstone for the development of civilization). [1] It is in the presence of humanity that will do everything to achieve perfection in many fields and has a record of achieving the greatest happiness in succeeding with the least amount of ignorance. There is no single method to solve all optimization problems, so many optimization methods have been developed to solve different types of problems. Optimization mathematically means the process of obtaining the minimum value or maximum value of a function with n variables f(x_1,x_2,…x_n). Most of the mathematical problems are presented in many practical applications that aim to find the optimal solution to a function, where x is a real dimensional vector n algorithms for solving mathematical optimization problems are classified into two types: deterministic algorithms. and Stochastic Algorithms. The first consists of algorithms based on the derivative as a prime factor of their work and are called algorithms that depend on the derivative (tendency). As for Stochastic algorithms, there are two types of algorithms, and although the difference between them is slight, they include: Heuristic Algorithms and Meta-Heuristic Algorithms [2].

The recent development of intuitive algorithms, or so-called meta-heuristic algorithms, and this term was first introduced by Glover in 1986, and in general, these algorithms work better than intuitive algorithms, in addition to that all meta-heuristic algorithms use confirmed swapping for random distribution and local search. It is worth noting This algorithm does not have a specific definition, and is sometimes known as the heuristic algorithm or Meta-heuristic, In general, it is a stochastic algorithm that relies on random distribution combined with local search [3]. Heuristic Algorithms Intuitive algorithms are modern algorithms that are characterized by speed and efficiency in finding the best ideal solution. These algorithms work to reduce the average search rate in order to find a solution to a problem. This type of algorithm is based on the local search method and is good for finding alternative solutions that work to achieve a correct balance. An example of this is the hill climbing algorithm [4]

Meta-heuristic Algorithms Post-intuitive algorithms are considered the most advanced and modern algorithms at the present time, which have been classified among the random algorithms, and they are powerful and flexible research methodologies that have succeeded in addressing many difficult scientific issues, they contain two main components: The first is Exploitation and the second is Diversification, meaning exploiting the search in a specific area and discovering possible solutions and choosing the best ideal solutions from among these solutions. Algorithms can be arranged on the basis of population or on the basis of society or on the basis of the path, so the Simulation Annealing Algorithms and the Tuba search Algorithms can be considered path-based algorithms, while the algorithms that depend on society include those inspired by nature and depend on the swarm, such as the Artificial Bee Algorithm, Ant Colony Algorithms, and Particle Swarm Optimization, The Arithmetic Optimization Algorithm as well as developmental algorithms such as the Genetic Algorithm [5, 6].

The Arithmetic Optimization Algorithm:

This algorithm is regarded as one of the post-intuitive approaches, as it can address optimization problems without requiring derivative calculations. It relies on basic arithmetic operations multiplication, division, subtraction, and addition while utilizing simple factors as part of the mathematical optimization process. The goal is to identify the optimal solution from a range of potential solutions. The algorithm is based on the following mathematical model:

2.1 Initialization Phase

The optimization process begins in the arithmetic optimization algorithm with a set of candidate solutions (x) as shown in the $matrix$ (x)

$$X = \begin{bmatrix} x_{11} & x_{12} \cdots & x_{1n} \\ x_{21} & x_{22} \cdots & x_{2n} \\ x_{N2} & x_{N2} \cdots & x_{N*n} \end{bmatrix} \quad (1)$$

Before we begin, we must define the research stage (exploration or exploitation). In this stage, MOA (Mathematical Optimization Accelerated) is used, which is a factor calculated by equation (2) and applied during the research stages. If r_1 > MOA, the system seeks to move away from the near-optimal solution. On the other hand, if r_2 < MOA, it converges towards the near-optimal solution.

$$MOA\ (C\_iter)=\text{min}+C_{iter} \times (\frac{max-min}{M_{iter}}) \qquad (2)$$

$MOA(C\_iter)$ indicates the value of the function at repetition t, and C_iter indicates the current repetition, and it ranges between or the maximum number of repetitions , M_iter and min, max, so it indicates the lower limit and the upper limit of the acceleration function, respectively [7].

2.2 Explorations Phase

This stage is base d on two main operations, either the division D or the multiplication M as in equation (3). The division D is considered the first rule in the equation and is conditional on $r2 < 0.5$ the last factor is neglected. Otherwise, the second factor M will participate to perform the current task instead of D

$$x_{i,j}\ (C_{iter}+1) = \begin{cases} best(x_j) \div (Mop{+}\in) \times (UBj - LBj) \times \mu + LBj) & r2 < 0.5 \\ best(x_j) \times Mop \times \left((UBj - LBj) \times \mu + LBj\right) & otherwise \end{cases} \qquad (3)$$

The random number is $r2$ and $x_{i,j}(C_{iter}+1)$ refers to the position j at the solution i of the iteration and the best position obtained so far is xj and ∈ is an integer Ubj. Ibj refers to the minimum The highest for the site μ in a control parameter that adjusts the search process and (Math Optimization Probability) refers to the value of the function for each iteration and is calculated by the following equation:

$$Mop\ (\ C_{iter}\ ) = 1 - \frac{C_{Iter}{}^{\frac{1}{\alpha}}}{M_{Iter}{}^{\frac{1}{\alpha}}} \qquad (4)$$

C_iter is the current iteration, M_iter is the maximum number of iterations, is α sensitive parameter [7]

2.3 Exploitation Phase

Through this stage, the optimal solution is discovered, which we can deduce in several fields using the operations subtraction (A) and addition (S) as shown in the equation (5), as the first rule in the equation is subtraction A conditional on r3 < 0.5, and the other factor (S) will be neglected until this factor finishes its current task, otherwise the second factor S will be involved to perform the current task instead of A [7].

$$x_{i,j}(C_{iter}+1) = \begin{cases} best(x_j\ ) - MOP \times \left((Ubj - Ibj) \times \mu + Ib_j\right) & r_3 > 0.5 \\ best(\ x_j) + MoP \times ((Ubj - Ibj) \times \mu + Ib_j) & otherwise \end{cases} \qquad (5)$$

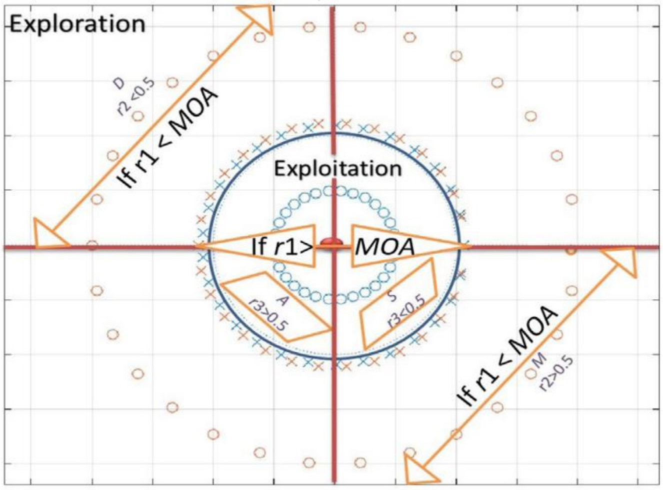

Fig. 1 *The Search phases of the AOA* [7]

Algorithm parameters AOA

$(r_1\ , r_2\ and\ r_3) \in [0, 1]$		
MOP_MAX =1	**C_Iter =1**	
MOP_MIN =0.2	**Alpha =5**	**Mu =0.499**

Algorithm 1 Pseudo-code of the AOA algorithm [7]

Initialize the Arithmetic Optimization Algorithm parameters α, μ.
Initialize the solutions‟ positions randomly. (Solutions: i=1, ..., N.)
while (C_ Iter < M _Iter) **do**
  Calculate the Fitness Function (F F) for the given solutions
  Find the best solution (Determined best so far).
    Update the MOA value using Eq. (2).
    Update the MOP value using Eq. (4).
    for (i=1 to Solutions) **do**
      for (j=1 to Positions) **do**
        Generate a random values between [0, 1] (r1, r2, and r3)
        if r1 >MOA **then**
        Exploration phase
        if r2 >0.5 **then**
          (1) Apply the Division math operator (D “ † ”).
          Update the ith solutions‟ positions using the first rule in Eq. (3).
        else
          (2) Apply the Multiplication math operator (M “ × ”).
          Update the ith solutions‟ positions using the second rule in Eq. (3).
        end if
        else
        Exploitation phase
        if r3 >0.5 **then**
          (1) Apply the Subtraction math operator (S “ − ”).
          Update the ith solutions‟ positions using the first rule in Eq. (5).
        else
          (2) Apply the Addition math operator (A “ + ”).

```
            Update the ith solutions‟ positions using the second rule in Eq. (5).
          end if
        end if
      end for
    end for
    C_ Iter=C_ Iter+1
end while
Return the best solution (x)
```

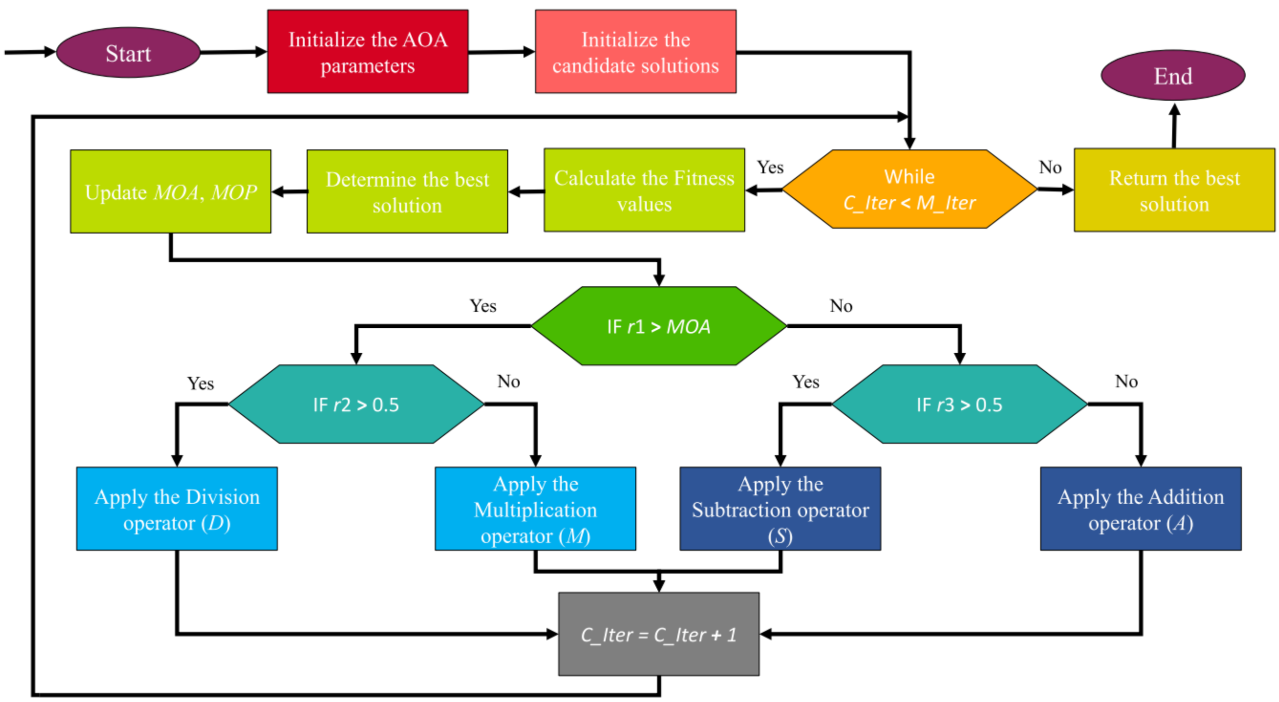

Fig. 2 *Flowchart of the proposed AOA*[7]

Conjugate Gradient Methods

The conjugate gradient method is one of the most important and popular methods for solving optimization problems that have a high scale in order to obtain the minimum value of the objective function, as it has advantages that fall between the regression method and the Newton method. It is faster than the steep regression method and it only requires calculating the derivatives of the first. It does not need to store and calculate the second derivatives that Newton's method requires, as it overcomes the slow convergence of this method as it does not require calculating the Hessian matrix or any of its approximations, this method is widely used to solve unconstrained optimization problems with large dimensions, as the gradient vectors are conjugate and linearly independent, as it works to minimize general convex functions or non-linear functions, and using the gradient direction It is the first step of the algorithm and plays an important role in reducing the function F(x) in the first iteration and that it converges when applied to solve non-quadratic problems.. [10-12].

Definition 1:

The optimization : it refers to identifying the optimal solution to a given problem by determining the minimum or maximum value of a function with n variables, where n is any integer greater than zero [12].

Definition 2: Global and local minimum

A. The global minimum value It signifies the minimum value of the function across the entire domain under investigation. [12].
B. Local minimum value: represents the lowest value of the function in specific locations within the search field. Algorithms that trend toward a global minimum are known as globally convergent algorithms, while algorithms that trend toward a local minimum are known as locally convergent algorithms.[12]

3.1 Derivation of new conjugation coefficients

Laylani et al. in 2023 proposed a parameterization of the conjugate gradient method [13]

$$d_{k+1}^{SBN} = \theta_k^{SBN}\, g_{k+1} + \frac{\|g_{k+1}\|^2}{\ell_k\, s_k\, y_k}\, d_k \qquad (6)$$

we have $\theta_k^{SBN} = \frac{d_k\, y_k^T}{\rho_k\, S_K^T\, y_k}$

The difference points are s_k and y_k derivatives difference

Let

$\ell_k = \alpha_k$ step size

$$d_{k+1}^{Dy} = -g_{k+1} + \beta^{Dy}\, d_k \qquad (7)$$

equation (6)= equation (7)

$$d_{k+1}^{Dy} = d_{k+1}^{SBN}$$

$$-g_{k+1} + \beta^{Dy}\, d_k = -\theta^{SBN}\, g_{k+1} + \frac{\| g_{k+1} \|^2}{\alpha_k\, S_K^T\, y_k}\, d_k] * y_k^T$$

Multiplying both sides of the equation by y_k^T we get

$$-y_k^T\, g_{k+1} + \beta^{Dy}\, y_k^T\, d_k = -\theta^{SBN}\, y_k^T\, g_{k+1} + \frac{\| g_{k+1} \|^2}{\alpha_k\, S_k^T\, y_k}\, y_k^T\, d_k$$

$$\beta^{Dy}\, y_k^T\, d_k = [1 - \theta^{SBN}]\, y_k^T\, g_{k+1} + \frac{\| g_{k+1} \|^2}{\alpha_k\, S_k^T\, y_k}\, y_k^T\, d_k$$

$$\frac{\| g_{k+1} \|^2}{y_k^T\, d_k}\, y_k^T\, d_k = [1 - \theta^{SBN}]\, y_k^T\, g_{k+1} + \frac{\| g_{k+1} \|^2}{\alpha_k\, S_k^T\, y_k}\, y_k^T\, d_k$$

By expanding Dy we get

$$\beta\tau = = \left[1 - \theta^{SBN}\right] y_k^T\, g_{k+1} + \frac{\|g_{k+1}\|^2}{\alpha_k\, S_k^T\, y_k} .\ y_k^T\, d_k$$

We have $S_k = \alpha_k d_k$ we get

$$\beta\tau = \left[1 - \theta^{SBN}\right] y_k^T\, g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2\, d_k^T\, y_k} .\ d_k^T\, y_k$$

$$\tau\beta = \left[1 - \frac{y_k^T\, d_k}{\alpha_k\ S_k\ y_k}\right] y_k^T\, g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2} .$$

Using also $S_k = \alpha_k\, d_k$

$$\tau\beta = \left[1 - \frac{1}{{\alpha_k}^2}\right] y_k^T\, g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}$$

$$\beta^{New} = \frac{\left[1-\frac{1}{{\alpha_k}^2}\right] y_k^T g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}}{\tau}$$

The sufficient slope property of the conjugate gradient method:

3.2 Theorem:

The search direction d_{k+1} defined in (8) is good for the characteristic convergent convergence algorithm to determine the value for all k values when the length of the Wolff characteristic matrix satisfies

$$g_{k+1}^T d_{k+1} \le -C \| g_{k+1} \|^2, \forall c > 0 \qquad (8)$$

Proof: Condition (1) will be proven using mathematical induction when $K = 0$

$$do = -g_0 \Rightarrow g_0^T do = -\| g_k \|^2 \qquad (9)$$

The theorem is correct. Now we assume that the theorem is true for all values of $k \ge 1$

$g_k^T d_k \le -C \| g_k \|^2, \forall c > 0 \qquad (10)$

We impose

$d_{k+1} = -g_{k+1} + \beta_k^{new} d_k \qquad (11)$

We multiply both sides of equation (11) by g_{k+1}^T

$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + \beta_k^{new} d_k g_{k+1}^T \qquad (12)$

When

$$\beta^{New} = \frac{\left[1-\frac{1}{{\alpha_k}^2}\right] y_k^T g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}}{\tau}$$

We substitute β^{New} into equation (12)

$$d_{k+1} = -g_{k+1} + \left[\frac{\left[1-\frac{1}{{\alpha_k}^2}\right] y_k^T g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}}{\tau}\right] d_k \qquad (13)$$

We multiply equation (13) by g_{k+1}^T

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + \left[\frac{\left[1-\frac{1}{{\alpha_k}^2}\right] y_k^T g_{k+1} + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}}{\tau}\right] d_k g_{k+1}^T$$

Using the Libashz continuity for the gradient difference from the Libashz continuity assumption we know that

$\| g_{k+1} - g_k \| \le l \| x_{k+1} - x_k \| = l \alpha_k \| d_k \|$

$\| y_k \| = \| g_{k+1} - g_k \| \le l \alpha_k \| d_k \|$

$y_k^T g_{k+1} \le l \alpha_k \| d_k \| \| g_{k+1} \| g_{k+1}$

We substitute this into the equation

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + \frac{1}{\tau}[\left[1-\frac{1}{{\alpha_k}^2}\right] l \alpha_k \| d_k \| \| g_{k+1} \| + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}] d_k g_{k+1}^T$$

From Wolff's Second Strong Condition, as shown in the equation below:

$g(x_k + \alpha_k d_k)^T d_k \le -\sigma g_k^T d_k \Rightarrow g_{k+1}^T d_k \le -\sigma g_k^T d_k$

$$g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 + \frac{1}{\tau} [\left[1-\frac{1}{{\alpha_k}^2}\right] l \alpha_k \| d_k \| g_{k+1} \|^2 + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}](-\sigma g_k^T d_k)$$

$$g_{k+1}^T\ d_{k+1} = -\parallel g_{k+1} \parallel^2 + \frac{1}{\tau}\left[\ l\ \alpha_k\ \parallel d_k \parallel d_k \parallel g_{k+1} \parallel^2 - \frac{1}{{\alpha_k}^2}\right](\sigma\ g_k^T\ d_k)$$

let $d_k = \frac{s_k}{\alpha_k}$

$$g_{k+1}^T\ d_{k+1} = -\parallel g_{k+1} \parallel^2 + \left[\frac{1}{\tau}\ l\ \alpha_k \frac{\parallel s_k \parallel}{\alpha_k} \parallel d_k \parallel\right] \parallel g_{k+1} \parallel^2$$

$$g_{k+1}^T\ d_{k+1} = -(1 - \frac{1}{\tau}[l \parallel s_k \parallel \parallel d_k \parallel] \parallel g_{k+1} \parallel^2$$

Let $l = 0.5$

$C = \left(1 - \frac{1}{\tau}[0.5 \parallel s_k \parallel \parallel d_k \parallel]\right)$

When $0 < c < 1$

Global convergence analysis of the conjugate gradient algorithm

Lemma (1)

Let us assume that Hypothesis (1) has been fulfilled, that the developed conjugate gradient method has been fulfilled, that d_k I s the direction of the slope search, and α_k has been established from the strong Wolff conditions (SWP) if

$$\sum_{k=1}^{\infty} \frac{1}{\parallel d_{k+1} \parallel^2} = \infty$$

$$lim_{k\to\infty}\ inf \parallel g_k \parallel = 0$$

Theorem(2)

We assume that hypothesis (1) has been fulfilled and that the conjugate gradient method has been fulfilled in the direction of the search slope d_k, and that the step length α_k has been generated from the conditions of (SWP).

$$lim_{k\to\infty}\ inf \parallel g_k \parallel = 0$$

Proof: Since the algorithm fulfills the theorem and that $g_{k+1} \neq 0$ we must prove $\parallel d_{k+1} \parallel$ constrained from above and taking $\parallel \ \parallel$ for both sides of equation (14)

$$d_{k+1} = -g_{k+1} + \beta_k\ d_k \qquad (14)$$

$d_{k+1} = \parallel -g_{k+1} + \beta_k\ d_k \parallel$ We take $\parallel . \parallel$ of equation (14)

$$\parallel d_{k+1} \parallel = \parallel -g_{k+1} \parallel + |\beta_k| \parallel d_k \parallel$$

$$\parallel d_{k+1} \parallel \leq \parallel -g_{k+1} \parallel + |\beta_k| \parallel d_k \parallel$$

$$\beta^{New} = \frac{\left[1 - \frac{1}{{\alpha_k}^2}\right] y_k^T\ g_{k+1} + \frac{\parallel g_{k+1} \parallel^2}{{\alpha_k}^2}}{\tau}$$

$$\parallel d_{k+1} \parallel = \parallel -g_{k+1} \parallel + \left| \frac{\left[1 - \frac{1}{{\alpha_k}^2}\right] y_k^T\ g_{k+1} + \frac{\parallel g_{k+1} \parallel^2}{{\alpha_k}^2}}{\tau} \right| \parallel d_k \parallel$$

$$\left|\beta^{New}\right| = \left| \frac{\left[1 - \frac{1}{{\alpha_k}^2}\right] y_k^T\ g_{k+1} + \frac{\parallel g_{k+1} \parallel^2}{{\alpha_k}^2}}{\tau} \right|$$

And using libishes condition

$$\parallel g_{k+1} - g_k \parallel \leq l \parallel x_{k+1} - x_k \parallel = l\ \alpha_k \parallel d_k \parallel$$

$$\parallel y_k = \parallel g_{k+1} - g_k \parallel \leq l\ \alpha_k \parallel d_k \parallel$$

and that

$$y_k^T\ g_{k+1} \leq l\ \alpha_k \parallel d_k \parallel \parallel g_{k+1} \parallel$$

We make up for it in β^{New}

$$\left|\beta^{New}\right| = \left| \frac{\left[1 - \frac{1}{{\alpha_k}^2}\right] l \ \alpha_k \ \| d_k \| \| g_{k+1} \| + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}}{\tau} \right|$$

Assume that

Q=$\| d_k \|$

R=$\| g_{k+1} \|$

$$\left|\beta^{New}\right| = \left| \frac{\left[1 - \frac{1}{{\alpha_k}^2}\right] l \ \alpha_k \ QR + \frac{\| g_{k+1} \|^2}{{\alpha_k}^2}}{\tau} \right|$$

$$\left|\beta^{New}\right| = N$$

$$\| d_{k+1} \| \leq \| g_{k+1} \| + N \ \| d_k \|$$

$$\| d_{k+1} \| \leq R + NQ \leq M$$

$$\| d_{k+1} \| \leq M$$

$$\frac{1}{\| d_{k+1} \|} \geq \frac{1}{M} \Rightarrow \sum_{k=1}^{\infty} \frac{1}{\| d_{k+1} \|^2} \geq \sum_{k=1}^{\infty} \frac{1}{M^2} = \frac{1}{M^2} \sum_{k=1}^{\infty} 1 = +\infty$$

$$lim_{k\to\infty} \ inf \ \| g_k \| = 0$$

Hybrid the Arithmetic Optimization Algorithm

In this section a modern method for solving fitness problems will be proposed will ,by proposing a new hybrid algorithm ,by linking the evolutionary ideas of the AOA algorithm with the traditional ideas of the conjugated gradient algorithm (CGA),called AOA-CG for short. The process is divided in each iteration into two stages .In the first stages randomness and velocity are used .for the AOA algorithm, and in the second stage ,the conjugated gradient algorithm and the beta value our use depends on hybridization using the developed conjugate gradient Which was developed by deriving a new parameter and then introducing the conjugate gradient algorithm into the arithmetic optimization algorithm.

Steps for hybridizing the arithmetic optimization algorithm with the conjugate gradient algorithm

Step 1: Initialize all primary X

Step 2: Developing the primary community using the developed CG

Step 3: Random initialization of the computational optimization algorithm parameters μ and α

Step 4: Specifically configure solution options (solutions N$\leq i \leq 1$)

Step 5:- While $(M\_(iter) > C\_iter)$ You do

Step 6: Calculate the value of the objective function (FF) for the solutions provided

Step 7: Finding the best solution (the best designed so far)

Step 8: Update the MON value using equation (4)

For (i=1 to Solution) it does

For (j=1 to Solution) it does

Step 9: For all solutions within Community X, perform the following steps:

- Generate random values between [0,1] $(r, r\_2 \ and \ r3)$
- $If \ MOA \ < \ r\_1$
- Exploration phase if $r2 > 0.5$

(1) Then apply the mathematics factor of section (D) and evaluate the two solutions using the first rule in equation (3).

(2) Finally apply the arithmetic multiplication operation $M"x"$ and update the solution modes using the first rule in equation (3)

- End then
- last
- Exploitation stage If r3 > 0.5 then

(3) Applying the mathematical subtraction operator (S"_")

(4) Update the solution positions using the second rule in equation (5)

- End then
- End then
- end
- end

Step 10: C_iter=C_iter+1

Step 11: The End of Time

Step 12: Return the best solution (X)

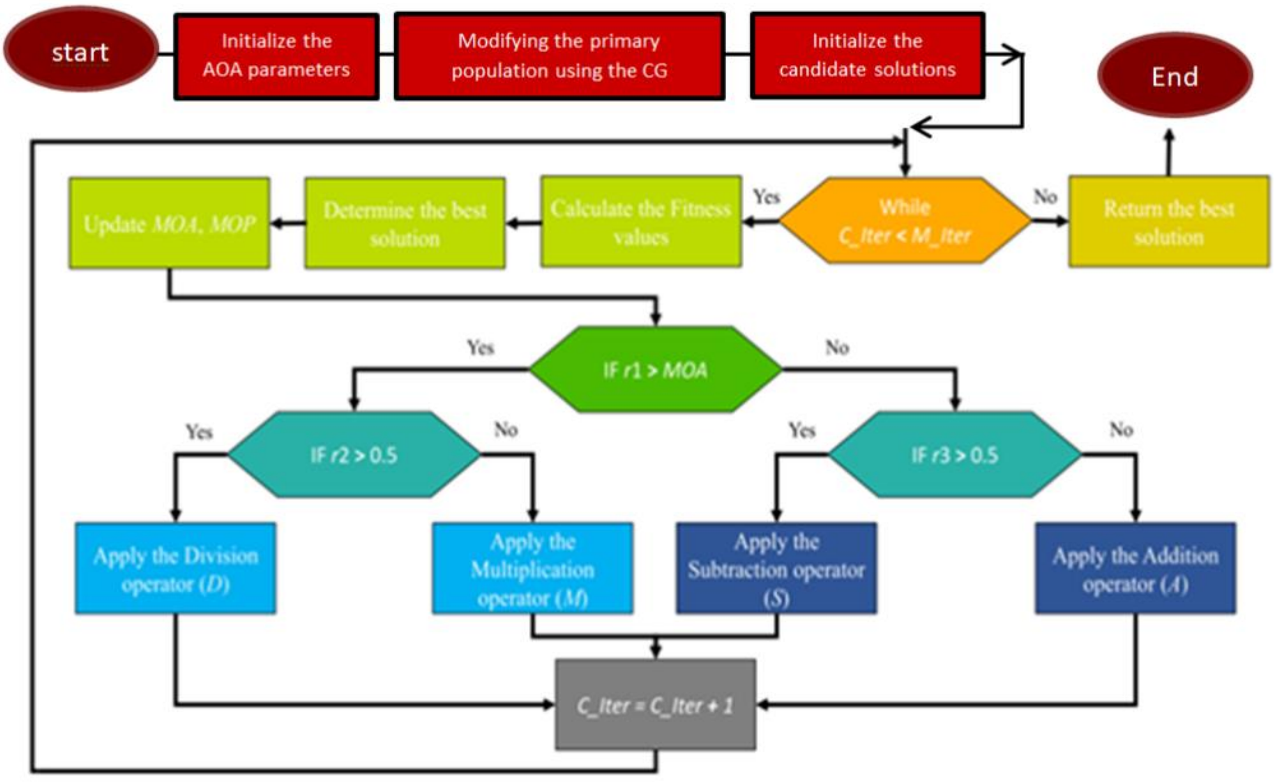

Fig. 3 *Flowchart of the arithmetic optimization algorithm with conjugate gradient algorithm AOA-CG*

Numerical Results

To estimate the efficiency of the proposed algorithms in finding the solution fitness difficulties, has been A0A algorithm-CG proposed test using 6 standard functions for comparison with the algorithm Arithmetic optimization themselves, Table 1displays the test information in addition to the specified range, the upper and lower limits for each function and the number of iterations. Where (500) ,(1000) repetitions and were used Which indicates the number of variables in the design table1. Details of test functions.

Table 1: Introducing standard benchmark test function (unimodal , multimodal) that is used to Assess the efficiency of algorithms [7]

Function	Description	dimensions	Range	$fmin$
F1	$f(x) = \sum_{i=1}^{n} 1x_i^2$	30,100,500,1000	[-100,100]	o
F2	$f(x) = \sum_{i=0}^{n} \|x_i\| + \prod_{i=0}^{n} \|x_i\|$	30,100,500,1000	[-10,10]	0
F3	$f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} x_j)^2$	30,100,500,1000	[-100,100]	0
F4	$f(x) = max_i \{\|x_i\|, 1 \leq i \leq n\}$	30,100,500,1000	[-100,100]	0
F5	$f(x) = \Sigma_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30,100,500,1000	[-5.12,5.12]	0
F6	$f(x) = 1 + \frac{1}{4000} \Sigma_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	30,100,500,1000	[-600,600]	0

As for the following table, it will explain the results of the original AOA algorithm and the results of the (AOA-CG) algorithm, By comparing the two algorithms, success is achieved. of the (AOA -CG) algorithm It is demonstrated by improved results. of the standard test functions of the high measurement, and obtaining ideal results for most of the mentioned functions, which confirms the success of the crossbreeding process.

Table 2: Comparison outcomes of AOA and AOA-CG using the number of 20 Elements and the number of Iterations (500) and (1000)

	AOA		**AOA-CG-B-NEW**	
Function	Iterations 500	Iterations 1000	Iterations 500	Iterations 1000
F1	5.5003×10^{7}	0	3.4543×10^{31}	0
F2	1.4746e$\times 10^{12}$	0	2.1683×10^{18}	0
F3	5.0708×10^{5}	0	1.4947×10^{28}	0
F4	0.00061474	6.8651×10^{261}	3.7203×10^{16}	0
F5	0	0	0	0
F6	0.1974896	0.098255	0	0

Discussion

We notice that the functions F1, F2, F3, F4, and F8 achieved the optimal (global) solution at iteration 1000, and the function F6 remained constant and did not cause any change, and the functions F5, F9 had very minor changes, and the function F7 did not cause any change, as the results of the hybrid algorithm were good. Very compared to the original algorithm, and through the hybrid algorithm we were able to reach the optimal solution in most functions by achieving the minimum values of those functions.

CONCLUSIONS

In this work, we enhanced the performance of the Arithmetic Optimization Algorithm (AOA) by hybridizing it with one of the most important conventional methods, the Conjugate Gradient (CG) method, for solving complex and large-scale optimization problems. This hybridization contributed to improving the performance of post-intuitive algorithms by increasing convergence speed, which in turn enhanced the quality of the resulting solutions and expanded exploration capabilities. The results of the AOA-CG algorithm were compared with the original AOA algorithm in solution (6) of the standard test functions. As shown in Table 2, the functions F1, F2, F3, F4, and F8 achieved the optimal (global) solution at iteration 1000, where the results of the hybrid algorithm were significantly better compared to the original algorithm. Through the hybrid algorithm, we were able to reach optimal solutions in most cases by achieving the minimum values of these functions.

References

1. W. I. Zangwill (1967). Non-linear programming via penalty functions. *Management science,* 13(5), 344-358, 1967.
2. X.-S. Yang (2010). Engineering optimization: an introduction with metaheuristic applications. *John Wiley & Son*s, USA.
3. T. E. Bruns & D. A. Tortorelli (2001). Topology optimization of non-linear elastic structures and compliant mechanisms, Computer methods in applied mechanics and engineering, 190(26-27), 3443-3459.
4. N.Kokash (2005). An introdaction to heuristic algorithms, *Dep.Informatics Telecommun*.
5. E.-G. Tallbi (2009). Metaheuristics: from design to implementation, John Wiley & Sons.
6. M. O. Okwu & L. K. Tartibu (2020). Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence Theory and Applications, *Springer Nature*.
7. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. & Gandomi, A.H. (2021). The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering, 376, 113609.
8. M. J. D. (1976). Powell, Some convergence properties of the conjugate gradient method, *Mathematical Programming*, 11(1), 42-49.
9. J. Nocedal & S. J. (2006). Wright, Conjugate gradient methods, Numerical optimization. Springer.
10. B. T. Polyak (1969). The conjugate gradient method in extremal problems, *USSR Computational Mathematics and Mathematical Physics*, 9(4), 94-112.
11. Antoniou, A. & Lu, W., (2007). Practical Optimization, algorithms and engineering application. *Springer Science& Business* Media, LLC. USA.
12. Beale, E.M.L. (1988). Introduction to Optimization. Wiley-Interscience*. Wiley–Blackwell*. USA.
13. Laylani, Y.A., Hassan, B.A. & Khudhur, H.M. (2023). Enhanced spectral conjugate gradient methods for unconstrained optimization. *Computer Science*, 18(2), 163-172.

IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS
مجلة سامراء للعلوم الصرفة والتطبيقية

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

الخوارزمية الحسابية المعدلة تعتمد على طريقة التدرج المترافق الجديدة

غفران ميسر[1]*، بان احمد[2]

1،2- قسم الرياضيات، كلية علوم الحاسبات والرياضيات، جامعة الموصل، العراق

البحث مستل من رسالة ماجستير الباحث الاول

معلومات البحث:

تاريخ الاستلام: 2024/11/25
تاريخ التعديل: 2024/12/29
تاريخ القبـــول: 2025/01/18
تاريخ الـنشر: 2025/06/30

الكلمات المفتاحية:

الامثلية،خوارزمية الامثلية الحسابية، طرائق التدرج المترافق، الخوارزمية الحدسية

معلومات المؤلف

الايميل: ghofran.23cspq4@student.uomosul.edu.iq

الموبايل: 07701788487

الخلاصة:

في هذا البحث، تم اقتراح خوارزمية هجينة جديدة، تجمع بين خوارزمية التحسين الحسابي (AOA) وهي خوارزمية وصفية تستخدم سلوك التوزيع للعمليات الحسابية الأساسية في الرياضيات، مثل الضرب (M)، والقسمة (D). والطرح (S)، والإضافة (A)، باستخدام خوارزمية التدرج المترافق الكلاسيكية (CGA). تُستخدم خصائص CGA لتعزيز المجموعة الأولية، والتي يتم إنشاؤها عشوائيًا باعتبارها المجموعة الأولية لخوارزمية AOA. نتائج الخوارزمية الهجينة أفضل بكثير من نتائج الخوارزمية الأصلية. ومن خلال هذا النهج المختلط، يتم التوصل إلى الحلول المثلى لمعظم الوظائف، مع الحصول على الحد الأدنى من القيم لهذه الوظائف. توضح المقارنة بين الخوارزميات الأصلية والهجينة أن الخوارزمية الهجينة تتفوق على الخوارزمية الأصلية. تم استخدام ست وظائف، مع إجراء مقارنات عند 500 و1000 تكرار.